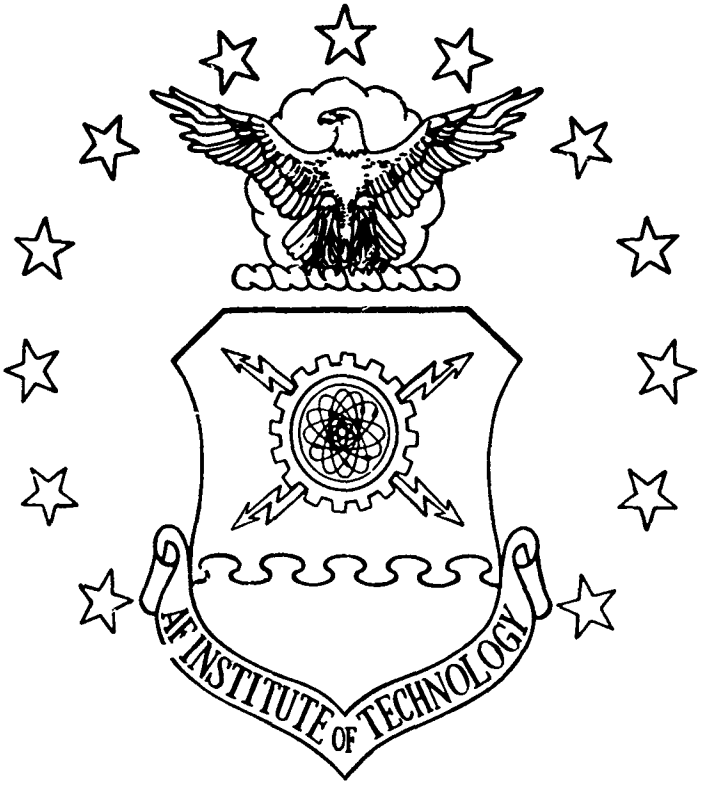
AD-A163 831



DTIC
ELECTE
FEB 1 0 1986
S D
D

A COMPARISON OF ESTIMATION TECHNIQUES FOR THE THREE PARAMETER PARETO DISTRIBUTION

THESIS

Dennis J. Charek
Major, USAF

AFIT/GSO/MA/85D-3

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC FILE COPY

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 2 10 010

AFIT/GSO/MA/85D-3

DTIC
ELECTE
FEB 1 0 1986
S D
D

A COMPARISON OF ESTIMATION TECHNIQUES FOR THE THREE PARAMETER PARETO DISTRIBUTION

THESIS

Dennis J. Charek
Major, USAF

AFIT/GSO/MA/85D-3

Approved for public release: distribution unlimited

AFIT/GSO/MA/85D-3

A COMPARISON OF ESTIMATION TECHNIQUES FOR THE THREE PARAMETER PARETO DISTRIBUTION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Space Operations

Dennis J. Charek, B.S., M.S.

Major, USAF

December 1985

Accesion For	
NTIS CRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

Approved for public release; distribution unlimited

QUALITY INSPECTED 3

Preface

The purpose of this research is to compare the minimum distance estimation technique with the best linear unbiased estimation technique to determine which estimator provides more accurate estimates of the underlying location and scale parameter values for a given Pareto distribution. Two forms of the Kolmogorov, Anderson-Darling, and Cramer-von Mises minimum distance estimators are tested. A Monte Carlo methodology is used to generate the Pareto random variates and the resulting estimates. A mean square error comparison is then performed to evaluate which estimator provides the best results. Additionally, various sample sizes and shape parameters are also used to determine whether they have an influence on a given estimator's performance.

I wish to express my sincere appreciation to my advisor, Dr. Albert H. Moore, for his guidance and direction throughout this thesis project. I also wish to thank my reader, Lt Col Joseph Coleman, for his advice and comments which greatly supplemented my efforts during this study. In addition, I would like to thank my classmate, Capt James Porter, for his assistance and contributions to this study.

Finally, I am very grateful to my wife, Monica, for her love, tolerance, and support throughout this thesis effort. I also wish to thank my daughter, Emily, and my son, Brian, for their understanding when playtime was interrupted by homework.

Dennis J. Charek

Table of Contents

	Page
Preface	ii
List of Figures	v
List of Tables	vi
Abstract	vii
I. Introduction	1
Specific Problem	4
Research Question	5
General Approach	5
Sequence of Presentation	5
II. Estimation Techniques	7
Estimation	7
Estimator Properties	9
Unbiased Estimators	9
Consistent Estimators	11
Efficient Estimators	12
Invariant Estimators	12
Summary	13
Empirical Distribution Function (EDF)	14
Best Linear Unbiased Estimator (BLUE)	15
Minimum Distance (MD) Estimator	16
Kolmogorov Distance	19
Cramer-von Mises Distance	20
Anderson-Darling Distance	20
III. Pareto Distribution	22
History	22
Applications	24
Socio-economic Related Aplications	24
Militarily Related Applications	25
Pareto Function	26
Grouping Pareto Distributions By Kind	27
Grouping Pareto Distributions By Parameter Number	28
Parameter Estimation	30
Best Linear Unbiased Estimator	31
BLUEs for Shape Greater Than 2	31
BLUEs for Shape Equal to or Less Than 2	36
Minimum Distance Estimator	38

IV. Monte Carlo Analysis 40

Monte Carlo Method 40
Monte Carlo Steps and Procedures 41
Step 1: Data Generation 41
Step 2: Estimate Computation 43
Step 3: Estimate Comparison 45

V. Results, Analysis and Conclusions 49

Results 49
Analysis 52
Conclusions 55

VI. Summary and Recommendations 56

Summary 56
Recommendations 58

Appendix A: Tables of Mean Square Errors 59

Appendix B: Computer Program for Estimator Comparison 70

Bibliography 105

Vita 109

List of Figures

Figure		Page
1.	Pseudocode for Program BLUMD	48
2.	Sample Table of Mean Square Errors	50

List of Tables

Table		Page
I.	Mean Square Error for c = 1 and n = 6	60
II.	Mean Square Error for c = 1 and n = 9	60
III.	Mean Square Error for c = 1 and n = 12	61
IV.	Mean Square Error for c = 1 and n = 15	61
V.	Mean Square Error for c = 1 and n = 18	62
VI.	Mean Square Error for c = 2 and n = 6	62
VII.	Mean Square Error for c = 2 and n = 9	63
VIII.	Mean Square Error for c = 2 and n = 12	63
IX.	Mean Square Error for c = 2 and n = 15	64
X.	Mean Square Error for c = 2 and n = 18	64
XI.	Mean Square Error for c = 3 and n = 6	65
XII.	Mean Square Error for c = 3 and n = 9	65
XIII.	Mean Square Error for c = 3 and n = 12	66
XIV.	Mean Square Error for c = 3 and n = 15	66
XV.	Mean Square Error for c = 3 and n = 18	67
XVI.	Mean Square Error for c = 4 and n = 6	67
XVII.	Mean Square Error for c = 4 and n = 9	68
XVIII.	Mean Square Error for c = 4 and n = 12	68
XIX.	Mean Square Error for c = 4 and n = 15	69
XX.	Mean Square Error for c = 4 and n = 18	69

Abstract

This investigation compared the minimum distance estimation technique with the best linear unbiased estimation technique to determine which technique provided more accurate estimates of the location and scale parameter values when applied to the three parameter Pareto distribution. Six distinct minimum distance estimators were developed. Of these six, two were based on the Kolmogorov distance, two were based on the Anderson-Darling distance, and two were based on the Cramer-von Mises distance. For a given sample size and Pareto shape parameter, the location and scale parameters were estimated. Additionally, varying combinations of sample sizes (6, 9, 12, 15, or 18) and shape parameters (1.0, 2.0, 3.0, or 4.0) were tested to investigate the affect of such changes.

A Monte Carlo methodology was used to generate the 1000 sample sets of Pareto random variates for each sample size - shape parameter combination with location and scale parameters both set to a value of 1. The best linear unbiased estimator and the six minimum distance estimators then provided parameter estimates based on the sample sets. Finally, these estimates were compared using the mean square error as the evaluation tool. The results of this investigation indicate that the best linear unbiased estimation technique provided more accurate estimates of location and scale for the three parameter Pareto distribution than did the minimum distance estimation techniques.

A COMPARISON OF ESTIMATION TECHNIQUES FOR THE THREE PARAMETER PARETO DISTRIBUTION

I. Introduction

Parameter estimation is an important underlying technique in statistical analysis. Although the statistician can perform some analysis intuitively, estimation requires a specific method. For example, if a statistician is asked to analyze some sample data, he could order it in ascending order and draw a histogram reflecting the occurrence frequency of values within certain intervals. Further, from the histogram's shape, he could guess the underlying population distribution. However, he could not easily determine the parameters (e.g. mean, standard deviation) of the population. At this point, the statistician needs a method to estimate the true population parameters from the sample data. The method is called the estimator, and the approximations based on the sample are the statistics (i.e. the estimates). Mendenhall defines an estimator as "a rule which specifically states how one may calculate the estimate based upon information contained in a sample" (23:13). Using these rules, the statistician can estimate the parameters of a population distribution based on sample data drawn from the population. These estimates then summarize the properties of the population for the investigator.

One estimation technique, called the best linear unbiased estimator (BLUE), relies on a linear combination of order statistics (10:265). Order statistics are a set of variables arranged according

to their magnitudes. For instance, ordering a set of observed random variables (e.g. fastest times in an automobile race) from smallest to largest results in a set of order statistics (24:229). The best linear unbiased estimator (T) can be used to estimate an unknown population parameter (θ) where T is only dependent on the values of n independent random variables. In addition, the estimator, T, must be linear in the set of n random variables. The estimator must also display the minimum variance among linear estimators and must be unbiased (10:265-266). In simple terms, unbiased means that on the average, the value of the estimator equals the parameter being estimated (33:197). Therefore, by combining a set of order statistics in a linear fashion, one can produce estimators for the underlying population parameters. If these estimators also possess the properties of minimum variance and unbiasedness, then they are called best linear unbiased estimators.

Another parameter estimation technique is minimum distance estimation, introduced by Wolfowitz in the 1950s as a method which "in a wide variety of cases, will furnish super consistent estimators even when classical methods...fail to give consistent estimators" (38:9). A minimum distance estimator is consistent if, as the sample size increases, the probability that the estimate approaches the true value of the parameter also increases (33:199). The minimum distance estimation technique is closely related in theory to the statistical procedure called goodness of fit because a distance measure is the evaluation criteria for both procedures. In goodness of fit, one tests the sample data to identify its underlying unknown distribution. A

goodness of fit test is "a test designed to compare the sample obtained with the type of sample one would expect from the hypothesized distribution to see if the hypothesized distribution function 'fits' the data in the sample" (8:189). Certain goodness of fit tests are based on a distance measure between the sample and a hypothesized distribution with known population parameters. Minimum distance estimation, however, reverses the goodness of fit approach by assuming a probability distribution type and then finding the values that minimize the distance measure. These values become the estimates of the population parameters (18:34).

Even though the minimum distance estimation technique was developed in 1953, researchers have not extensively studied the technique until recently. Parr and Schucany reported in 1979 that the method yields "strongly consistent estimators with excellent robustness properties" (27:5) when used to estimate the location parameter of symmetric distributions (27:5). Robustness of an estimator is its ability to serve as a good estimator even when the distribution assumptions are not strictly followed (27:3). Additionally, several Air Force Institute of Technology (AFIT) students, under the guidance of Dr. Albert H. Moore, have completed thesis research projects by applying the minimum distance estimation technique to specific distributions and comparing this technique with other estimation methods. These former students include Maj McNeese, working with the generalized exponential power distribution; Capt Daniels, working with the generalized t distribution; Capt Miller, working with the three parameter Weibull distribution; Capt James, working with the three parameter gamma distribution; 2Lt

Bertrand, working with the four parameter beta distribution; and 2Lt Keffer, working with the three parameter lognormal distribution. Results have generally shown that minimum distance estimators provide better estimates (i.e., estimates closer to the actual population parameters) than the other techniques used (4:9).

The literature search reveals that the capabilities of the minimum distance estimation technique have not been compared with those of the best linear unbiased estimator with regard to the Pareto distribution, a distribution of considerable value. The Pareto distribution has a variety of uses in the commercial sector. Johnson and Kotz identify several Pareto distribution analysis areas, including city population distribution, stock price fluctuation, and oil field location (16:242). In addition to commercial users, the Air Force also uses the Pareto distribution in a number of analysis areas: time to failure of equipment components (9), maintenance service times (14), nuclear fallout particles' distribution (11), and error clusters in communications circuits (3). In sum, the Pareto distribution proves to be a distribution worthy of further investigation. Use of the minimum distance estimation technique applied to the Pareto distribution offers the researcher a chance to expand the frontier of knowledge in this area.

SPECIFIC PROBLEM

Researchers have not explored the potential of the minimum distance estimation technique to improve upon the best linear unbiased estimation technique as applied to the Pareto distribution. A comparison of the techniques in a controlled environment is needed to evaluate which

technique performs better under given circumstances. The controlled environment should specify the sample size and the value of the parameters of the underlying Pareto distribution function for each comparison attempt.

RESEARCH QUESTION

For specified parameter values and sample sizes, which estimation technique, minimum distance or best linear unbiased, performs better when applied to the Pareto distribution?

GENERAL APPROACH

Monte Carlo analysis is the analytical method to be used to make the estimation technique comparison. Monte Carlo analysis of estimation methods consists of three steps. First, one generates random variates from a specified Pareto distribution (i.e., a Pareto distribution with known parameters). Second, the two estimation techniques are used to obtain parameter estimates based on the random sample data from the first step. Third, the resulting estimates are compared to determine which estimation technique provided the better parameter estimates (4:27). The mean square error technique can be used to perform this evaluation (4:31).

SEQUENCE OF PRESENTATION

This report will proceed with five additional chapters. The second chapter will discuss the estimation techniques used in this study while the third chapter will present the Pareto Distribution. The fourth chapter will describe the Monte Carlo analysis methodology used to make

the estimation technique comparisons. The fifth chapter will present the results and conclusions of the study while the sixth chapter will provide a short summary and some recommendations for future study in this area.

II. Estimation Techniques

This chapter will first provide a discussion on estimation in general, some desirable properties of estimators, and the empirical distribution as an estimator of the true distribution. Following this discussion, the two estimation techniques to be compared in this thesis will be presented. First the best linear unbiased technique will be discussed along with its inherent properties. Then the minimum distance technique will be presented in the three distance measure forms to be used throughout the rest of this study.

ESTIMATION

Estimation is part of a larger area of study called statistical inference. The statistician makes inferences about the state of nature, or the "way things really are" (22:187), based on data gathered from experiments done to discover something about the state of nature (22:187). Lindgren then narrows his discussion of statistical problems to decision problems, eliminating the areas of experimental design and representative data gathering.

> Some statistical problems, notably in business and industry, are decision problems, in which the partial information about the state of nature provided by data from experimentation is used as the basis of making an immediate decision [22:188].

Lindgren then describes the general decision problem as consisting of "a set or 'space' A of possible actions that might be taken, the individual 'points' of this space being the individual actions" (22:188). He finally defines estimation problems as "those in which the action space A is identical with the space of parameter values that index the family

of possible states of nature" (22:188). In this case, states of nature could be described by the distribution function family members, each member being defined through its own set of parameter values.

Pritsker describes the concept of parameter estimation by presenting two supporting definitions. He first defines the 'population' as the set of data points consisting "of all possible observations of a random variable" (31:46). He then defines a 'sample' as being "only part of these observations" (31:46). A method to summarize a set of data is "to view the data as a sample which is then used to estimate the parameters of the parent or underlying population" (31:46). Runyon and Haber simply define a parameter as "a summary numerical value calculated from a population" (33:4).

Liebelt indicates that the estimation problem, defined earlier by Lindgren, is difficult to solve. In fact, because there can be many estimates regarding a problem, the solution is not unique. Therefore, the statistician begins searching for the 'best' estimate; but, since the criteria for a 'best' estimate is arbitrary, there cannot be an optimal estimate to solve all problems (21:135-136). "Each problem may require a different set of optimal criteria; the choice is always left to the user of estimation theory" (21:136). So, the search always continues for a better estimator. This thesis is a continuation of that search.

Before we continue by listing and defining some of the agreed upon properties of a good estimator, we must clarify the difference between an estimator and an estimate. Mendenhall explains that an estimator is "a rule which specifically states how one may calculate the estimate based upon information contained in a sample" (23:13). However, when

the estimator is used to produce a particular value based on specified sample data, "the resulting predicted value is called an estimate" (23:13). Wine draws an analogy to describe the difference. He indicates the distinction between the two is the same as the difference between a function, f(x), and the evaluated functional value, f(c). "f(x) is a variable defined in some domain of x, and f(c) is a constant corresponding to a specified value of x equal to constant c" (37:170-171). Before a sample is drawn, we have an estimator. After the sample is drawn, the estimator produces a particular value which is an estimate (37:171).

ESTIMATOR PROPERTIES

The search for better estimators continues; but, what is the criteria for determining a good estimator? Certain properties of estimators have been defined and seem to be reasonable guides for choosing good estimators, although these criteria cannot be fully "justified except on the basis of intuition" (21:136). This section will discuss four of these desirable properties. If an estimator is to be used in repeated samplings from the same population, then unbiasedness is a desirable property; otherwise, a biased estimator could possibly be found which provides better parameter estimates. Additionally, a good estimator should be consistent, efficient and invariant. Each of these properties will now be described in more detail.

Unbiased Estimators. The first property a good estimator to be used in repeated samplings from the same population is unbiasedness. Freeman defines an unbiased estimator as follows:

> We have a population described by the density function $f(x;\theta)$, where f is known and the value of the parameter is unknown. A random sample $x_1, x_2, \ldots, x_n$ is drawn from this population. The statistic $t(x_1, x_2, \ldots, x_n)$ is an unbiased estimator of the parameter θ if
>
> $$E(t) = \theta \tag{2.1}$$
>
> for all n and for any possible value of θ [10:229].

Wine points out that this definition "requires that the mean of the sampling distribution of any statistic equals the parameter which the statistic is supposed to estimate" (37:172). In other words, the expected value of the statistic t equals the parameter being estimated, where "the expected value of a random variable x with density function f(v) is defined as

$$E(x) = \int_{-\infty}^{\infty} v\ f(v)dv \tag{2.2}$$

(21:85). Freeman defines the term density function as "a function $f(x_i)$ which is connected to probability statements on the random variable x by

$$p(x = x_i) = f(x_i) \tag{2.3}$$

(10:18). Looking at unbiasedness from a slightly different perspective, Liebelt says that unbiasedness "is desirable, for it states that in the absence of measurement error, and uncertainty in the estimation procedure, the estimate becomes the true value" (21:137). Freeman adds a final note concerning unbiased estimators. He indicates that for an estimator to be truly unbiased, Eq (2.1) "is required to hold for all sample sizes n" (10:229). There are cases when Eq (2.1) roughly holds

only for very large sample sizes. In these cases, the estimator is merely 'asymptotically unbiased' (10:229).

Unbiasedness is an important property for an estimator to have in repeated samplings from the same population. The reason for this statement becomes apparent when one looks at what can happen if an estimator is biased. "Any estimating process used repeatedly and which on the average (mean) is not equal to the parameter leads to a sure cumulation of error in one direction" (10:229). To avoid this accumulation of error in one direction, the statistician seeks to find and use unbiased estimators. However, in a single estimation situation, unbiasedness may not be desireable. Instead, one could seek to minimize the mean square error of the estimate which could then result in a better estimate.

Consistent Estimators. The second property of a good estimator is that of consistency. As the sample size increases, one would want the risk associated with the estimator to decrease. "That is, the estimator ought to be better when it is based on twenty observations than when it is based on two observations" (25:172). This supposition portrays the idea of consistency. "An estimator is consistent if for a large sample there is a high probability that the estimator will be near the parameter it is intended to estimate" (5:140).

A similar definition expressed by Wine uses the idea of convergence to define a consistent estimator. An estimator, t, of the parameter θ is consistent if, for any small numbers δ and ϵ, "there exists an integer n' such that the probability that $[|t - \theta| < \epsilon]$ is greater than $[1-\delta]$ for all $n > n'$ " (37:171). This definition introduces the idea of convergence by saying, "given any

small [ê], we can find a sample size large enough so that, for all larger sample sizes, the probability that [t] differs from the true value θ [by] more than ê is as small as we please" (37:171). Therefore, the estimator, t, converges in probability to θ (37:171). Consistency, then, implies that as sample sizes increase, the probability also increases that the estimator provides estimates which more closely approximate the true value of the parameter being estimated.

Efficient Estimators. The third desirable property of a good estimator is that of efficiency. Efficiency is generally used as a measure to compare two estimators. The efficiency is the ratio of their mean square errors. Mendenhall and Scheaffer indicate that the mean square error can be written as the summation of the variance and the square of the bias of an estimator (24:267).

Since variance is a measure of the dispersion of the distribution of an estimator about the parameter value, the statistician seeks an estimator with small variance. By selecting an estimator with the smaller variance, he ensures that his estimates will converge more rapidly to the true parameter value (32:155). Therefore, "one estimator is said to be more efficient than another when the variability of its sampling distribution is less" (33:198).

Invariant Estimators. The final property of a good estimator is that of invariance. Invariance is particularly desirable when functional transformations must be made regarding the parameter. As Freeman states:

> We call a method of estimation invariant under transformation of a parameter if, when the method leads to t as the estimator of θ, the method also leads to $g(t)$ as the estimator of $g(\theta)$. We can speak of t as an invariant estimator for a certain class of transformations g if, when the parameter θ is transformed by g to $g(\theta)$, the estimator t is transformed to $g(t)$ [10:233].

If the statistician is working with an invariant estimator where the estimate of θ is t, then he can conclude that his estimate for $\theta + k$ is $t + k$ and his estimate for $k\theta$ is kt (10:233). Thus, the property of invariance permits the transformation of a parameter to be translated into the transformation of its estimator.

Summary. Three desirable properties of an estimator are consistency, efficiency, and invariance. Unbiasedness is desirable when the estimator is used in repeated sampling from the same population. Unbiasedness means that, on the average, the estimator equals the parameter being estimated. Consistency means that as the sample size increases, the estimator will more closely approximate the true parameter value. Efficiency is a comparative measure between estimators where the estimator with the smaller mean square error is more efficient. Finally, invariance means that if a transformation operation is performed on a parameter, the identical transformation can be performed on the estimator resulting in the transformed estimator becoming a valid estimator for the transformed parameter. Although these properties are desirable, estimators generally do not possess all of these properties. Therefore, the statisticians must find an estimator with the properties needed for their particular applications.

EMPIRICAL DISTRIBUTION FUNCTION (EDF)

An empirical distribution is a distribution based solely on sample values of a random variable. The empirical distribution can be thought of as an estimation of the true underlying population distribution. The empirical distribution is developed "by observing several values of the random variable and constructing a graph S(x) that may be used as an estimate of the entire unknown distribution function F(x) of the random variable (8:59). Conover defines the empirical distribution as follows:

> Let $X_1, X_2, \ldots, X_n$ be a random sample. The empirical distribution function S(x) is a function of x, which equals the fraction of X_is that are less than or equal to x for each x, $-\infty < x < \infty$ [8:69].

Based on this definition, the graph of the empirical distribution function, S(x), is a step function starting at zero. As each sample value (ordered from lowest to highest) is encountered, a step of height 1/n is entered on the graph. This procedure continues until all the sample values have been entered and a height of one has been reached. "S(x) resembles a distribution function in that it is a nondecreasing function that goes from zero to one in height. However, S(x) is empirically (from a sample) determined and therefore its name" (8:70). The empirical distribution function is used as an estimator for the population distribution function of the random variable (8:70).

From the empirical distirbution function, one can "compute the expectation of the empiric random variable, E(x). We have

$$E(x) = \sum_{i=1}^{n} x_i(1/n) = (1/n) \sum_{i=1}^{n} x_i \qquad (2.4)$$

which is just the sample mean, $\bar{x}$ (5:137). Eq (2.4) uses the

discrete random variable form of the expected value definition. Therefore, assuming the empirical distribution acceptably estimates the population distribution leads to the sample mean being an acceptable estimate for the population mean (5:138).

BEST LINEAR UNBIASED ESTIMATOR (BLUE)

Knowing what properties are desirable in an estimator still leaves the statistician with the problem of developing an estimator. One estimator is called the best linear unbiased estimation technique. As was mentioned in Chapter I, the BLU estimator is based on order statistics, which is simply an arrangement of random variables in order of magnitude (24:229). A population parameter (θ) can be estimated by a statistic (T) which depends only on the values of n independent random variables: $x_1, x_2, \ldots, x_n$ (10:265).

The title of this estimator indicates some of the properties that it possesses. Namely, the estimator must be unbiased, 'best', and linear. As was discussed ealier in this chapter, an unbiased estimator has a bias term equal to zero, and, on the average over many trials, the estimator provides estimates equal to the parameter value. Eq 2.1 states the property mathematically. In addition to being unbiased, the BLU estimator must be 'best'. To be best among unbiased estimators, the estimator must have the minimum mean square error (10:265). The mean square error is the sum of the variance term and the square of the bias term (24:267). Since we are dealing with an estimator which is inherently unbiased (i.e., the bias term equals zero), the mean square error simply reduces to the variance term. Therefore, in this case, best implies minimum variance. Finally, the

BLU estimator must be linear. Linearity demands that we consider only "estimators which are linear in the random variables $x_1, \ldots, x_n$, for it is only in comparison with other esimators within this restricted class that we can always find estimators [which are best unbiased]" (10:266). Stated mathematically, the estimator appears as follows:

$$T = c_1 x_1 + \ldots + c_n x_n \qquad (2.5)$$

where the coefficients (c_i) must be determined (10:266).

In addition to the properties described above, the best linear unbiased estimator possesses another desirable feature, that of invariance. Mood and Graybill indicate that BLU estimators are a subset of least-squares estimators (25:349). Further, they state that, in general, least square estimators do not possess the invariance property. "There is one important case, however, when the invariant property holds for least-squares estimators, and this is the case of linear functions" (25:350). Therefore, in addition to being unbiased, and possessing minimum variance, the BLU estimator is also invariant.

MINIMUM DISTANCE (MD) ESTIMATOR

Chapter I presented a partial history and description of the minimum distance estimation technique. The efforts of Wolfowitz culminated in his 1957 paper which refined his work toward "developing the minimum distance method for obtaining strongly consistent estimators (i.e., estimators which converge with probability one)" (39:75). In the paper, he emphasized that his method could be used with a variety of distance measuring techniques (39:75). Additionally, Wolfowitz stated

that "it is a problem of great interest to decide which, if any, definition of distance yields estimators preferable in some sense" (39:76). This thesis will in part respond to this challenge, since three distance measures will be used in the minimum distance method for comparison against the best linear unbiased estimation method. The three distance measures to be used are the Kolmogorov, the Anderson-Darling, and the Cramer-von Mises discrepancy measures. Wolfowitz finally summarizes the minimum distance method as follows:

> The estimator is chosen to be such a function of the observed chance variables that the d.f. of the observed chance variables (when the estimator is put in place of the parameters and distributions being estimated) is 'closest' to the empiric d.f. of the observed chance variables [39:76].

Since 1957, the minimum distance estimation technique has been studied by many other statisticians and has been found to display other desirable estimator properties. The technique has "been considered as a method for deriving robust estimators by Knusel (1969) and Parr and Schucany (1980)" (28:178). Additionally, Parr and Schucany indicate that the method yields "strongly consistent estimators with excellent robustness properties" (27:5) when used to estimate the location parameter of symmetric distributions (27:5). They define robust estimation as "efficient or nearly efficient (at a model) estimation procedures which also perform well under moderate deviations from that model" (27:2). They attempt to explain why the minimum distance estimator possesses robustness properties:

> It may well be inquired as to why an estimator obtained by minimization of a discrepancy measure which is useful for goodness-of-fit purposes (and, hence, in many cases extremely sensitive to outliers or general discrepancies from the model) should be hoped to possess any desirable 'robustness' properties. It turns out that, in most

> cases . . . while the discrepancy measure itself may be fairly sensitive to the presence of outliers, the value . . . which minimizes the discrepancy . . . is much less so [27:5-6].

Finally, they state that the method presents a trade-off between efficiency considerations and robustness considerations (28:179).

In addition to consistency, robustness and efficiency, investigators have revealed other attractive features of the minimum distance estimation technique. Parr and Schucany indicate that "minimum distance estimators share an invariance property with maximum likelihood estimators . . . It operates in a manner analogous to maximum likelihood methods in simply selecting a 'best approximating distribution' from those in the model" (27:9). Additionally, Parr states that the method is very easy to implement. "Given a set of data, a parmetric model, and a distance measure between distribution functions, all that is needed is an omnibus minimization routine to compute the estimator" (26:1207-1208). Finally, minimum distance estimators provide meaningful results even if the conjectured parametric model is incorrect. MD-estimation still provides the best approximation in terms of probability units with regard to the conjectured distribution (26:1208). "This is a feature not enjoyed by other estimation methods such as the maximum likelihood" (26:1208). Therefore, MD-estimation can be a very useful tool for the statistician.

The minimum distance estimation technique uses a distance measure and, for this reason, is closely linked with certain goodness-of-fit tests. As explained by Stephens, goodness-of-fit statistics are "based on a comparison of F(x) with the empirical distribution function $F_n(x)$" (35:730). In a goodness-of-fit test, one is interested in fitting an empirical distribution function, described

earlier, with a fully specified (i.e., with known paramters) distribution function. The test for whether the fit is 'good' is normally a measure of distance between the two distribution curves. In contrast, minimum distance estimation uses a parent distribution family with certain unknown parameters. The estimates of the unknown parameters are those parameter values which minimize the distance measure between the empirical distribution and the parent distribution being investigated. The three distance measures to be used in this study are described next.

Kolmogorov Distance. The statistic suggested by Kolmogorov in 1933 is the largest absolute distance between the graphs of the empirical distribution function, $S(x)$, and the hypothesized distribution function, $F(x_i;\theta)$ measured in the vertical direction (8:345). Symbolically, the Kolmogorov distance (D) is given by:

$$D = \sup|F(x_i;\theta) - S(x)| \tag{2.6}$$

which reads D equals "the supremum, over all x, of the absolute value of the difference $F(x_i;\theta) - S(x)$ " (8:347). Stephens provides a computational form for all of the distance measures to be used in this study where he lets $z_i = F(x_i)$, $i = 1,2,\ldots,n$. For the Kolmogorov distance, the computational form is as follows:

$$D^+ = \max_{1\leq i\leq n}[(i/n) - z_i]$$

$$D^- = \max_{1\leq i\leq n}[z_i - (i-1)/n]$$

$$D = \max(D^+, D^-) \tag{2.7}$$

(35:731). These computational formulae provide the maximum distance

between the empirical distribution function, which is a step function, and the conjectured distribution function, $F(x_i;\theta)$.

Cramer-von Mises Distance. The Cramer-von Mises statistic is actually a member of the Cramer-von Mises family of distance measures which is "based on the squared integral of the difference between the EDF and the distribution tested:

$$W^2 = \int_{-\infty}^{\infty} \{F_n(x) - F(x;\theta)\}^2 \beta(x)\, dx \qquad (2.8)$$

The function . . . [$\beta(x)$] . . . gives a weighting to the squared difference" (34:2). The Cramer-von Mises statistic is produced by setting the weighting function equal to one, $\beta(x) = 1$ (34:2). The computational form of the Cramer-von Mises statistic is given by Stephens as follows:

$$W^2 = \sum_{i=1}^{n} [z_i - (2i - 1)/2n]^2 + (1/12n) \qquad (2.9)$$

(35:73i). This formula uses the same symbology as the computational form of the Kolmogorov distance measure.

Anderson-Darling Distance. The Anderson-Darling distance measure is actually another member of the Cramer-von Mises family. In this case, however, the weighting factor is $1/\{u(1 - u)\}$ where $0 < u < 1$ (27:4). "This weight function counteracts the fact that the discrepancy [in Eq 2.9] between $F_n(x)$ and $F(x;\theta)$ is necessarily becoming smaller in the tails, since both approach 0 and 1 at the extremes (34:2). Therefore, the Anderson-Darling weighting function gives "greater importance to observations in the tail than do most of

the EDF statistics" (34:2). Stephens gives the computational form of the Anderson-Darling statistic as follows:

$$A^2 = -\{\sum_{i=1}^{n} (2i - 1)[\ln z_i + \ln (1 - z_{n+1-i})]\}/n - n \qquad (2.10)$$

(35:731). Again, this computational formula uses the same symbology used for the other two distance measures' computational formulae.

III. Pareto Distribution

This chapter will first relate the history of the Pareto distribution. A summary of various socio-economic and military applications will follow this historical perspective. Then a detailed description of the Pareto function will be presented. Finally, this chapter will describe the best linear unbiased and the minimum distance estimation techniques as applied specifically to the Pareto function.

HISTORY

In 1897 Vilfredo Pareto (1848-1923), an Italian-born Swiss professor of economics, formulated an empirical law which bears his name (16:233). Pareto's Law was based on his study of the distribution of incomes in several European countries during the nineteenth century. The mathematical results of the study were summarized as follows:

$$N = Ax^{-c} \tag{3.1}$$

where N is the number of people haveing incomes equal to or greater than income level x. A and c are parameters where c is sometimes referred to as Pareto's constant or the shape parameter (16:233). Pigou summarized Pareto's findings in the following statement:

> It is shown that, if x signify a given income and N the number of persons with incomes exceeding x, and if a curve be drawn, of which the ordinates are logarithms of x and the abscissae logarithms of N, this curve, for all the countries examined, is approximately a straight line, and is, furthermore, inclined to the vertical axis at an angle, which, in no country, differs by more than three or four degrees from 56°. This means (since tan 56° = 1.5) that, if the number of incomes greater than x is equal to N, the number greater than mx is equal to $[\ N(1/m)^{1.5}\ ]$,

> whatever the value of m may be. Thus the scheme of income distribution is everywhere the same [29:647].

The Pareto premise, then, as deduced from his mathematical findings and stated in economic rather than mathematical terms is as follows:

> Hence, what this thesis amounts to in effect is that, on the one hand, anything that increases the national dividend must, in general, increase also the absolute share of the poor, and, on the other hand--and this is the side of it that is relevant here--that it is impossible for the absolute share of the poor to be increased by any cause which does not at the same time increase the national dividend as a whole . . . we cannot be confronted with any proposal the adoption of which would both make the dividend larger and the absolute share of the poor smaller, or 'vice versa' [29:648].

Pareto felt, therefore that his law was "universal and inevitable--regardless of taxation and social and political conditions" (16:233).

Since the statement of Pareto's Law, several renowned economists have refuted the law's sweeping applicability (16:233). In particular, Pigou identified defects in its statistical basis, arguing that the differences in inclination of the plotted lines were significant. Additionally, he argues that such a generalization from an empirical study under certain conditions (certain avenues of income such as inheritance and personal effort) cannot justifiably be extended to all social conditions (29:649-655).

> The general defence of "Pareto's Law" as a law of even limited necessity rapidly crumbles. His statistics warrant no inference as to the effect on distribution of the introduction of any cause that is not already present in approximately equivalent form in at least one of the communities--and they are very limited in range--from which these statistics are drawn. This consideration is really fatal; and Pareto is driven, in effect, to abandon the whole claim . . . [29:654-655].

Additionally, Champernowne identifies weaknesses in the Pareto Law.

He indicates that the use of the Pareto constant as a measure of income distribution inequality between communities suffers from two problems. Firstly, the measure only addresses income before taxation. Secondly, the measure only applied to income distributions among the rich and breaks down when applied to those with medium incomes (7:609).

Finally, Fisk discusses the value of the Pareto distribution regarding its ability to describe distributions of income. He states that the "Pareto curve fits income distributions at the extremities of the income range but provides a poor fit over the whole income range" (12:171).

Therefore, Pareto's Law with regard to income distributions is no longer highly touted. However, other disciplines have found application of the Pareto distribution to be very useful.

APPLICATIONS

Socio-economic Related Applications. Although the Pareto distribution was formulated as a reflection of income distribution, the Pareto distribution has proven to be useful in many other areas of investigation. Johnson and Kotz indicate the Pareto distribution can be useful in describing many socio-economic or naturally occuring quantities. Examples include the distributions of city population sizes, fluctuations in the stock market, and the occurrence of natural resources. The Pareto is useful in these areas because they often display statistical distributions with very long right tails (16:242).

Koutrouvelis listed some additional areas where the Pareto distribution had successfully been used. These areas include: business mortality rates, worker migration, property values and inheritance, and

service times in queues (19:7).

Johnson and Kotz additionally identified the area of personal income investigation as an area where the Pareto distribution was applicable (16:242). In 1982, Wong used the Pareto in his analysis of income. He indicates that many individuals underreport their true incomes to avoid a portion of their tax payments. Wong shows the applicability of the Pareto in reflecting this underreporting phenomena (40:1).

Militarily Related Applications. In addition to socio-economic interests, the Pareto distribution has proven useful in many areas of interest to the military. These areas include fallout mass-size distributions, interarrival time distributions, and failure time distributions. This section will address each of these areas in turn.

E. C. Freiling conducted a study for the U.S. Naval Radiological Defense Laboratory concerning a comparison of distribution types for describing "the size distribution of particle mass in the fallout from land-surface bursts" (11:1). In this study, he compared the lognormal distribution with the Pareto. He determined that with the effects of the uncertainties playing in the problem, the differences in descriptive ability of the two distributions were trivial. He indicated that the lognormal "has the esthetic advantage of an observationally confirmed theoretical basis in the case of airburst debris" (11:12). However, if truncation is required, the Pareto distribution has "the practical advantage of simplifying further calculations of particle surface distribution" (11:12).

A Pareto description of interarrival times has played an important part in two other studies, one involving interarrival times in general

and the second involving telephone circuit error clustering. Bell, Ahmad, Park and Lui performed the general interarrival time study supported by a grant from the Office of Naval Research. They indicate that interarrival time distributions are usually thick-tailed as compared to Gaussian or Poisson processes for like distributions. They state that the Pareto can provide a variety of tail thicknesses depending on the value of the shape parameter employed (2:1). In the telephone circuit paper, Berger and Mandelbrot propose a new model to describe error occurrence on telephone lines. They conclude that the Pareto distribution can well be used to approximate the distribution of inter-error intervals.

Finally, the Pareto distribution has proven useful in life testing and replacement policy situations. Davis and Feldstein show the Pareto as a competitor to the Weibull distribution with regard to time to failure of a system since, "unlike the Weibull, it does not give rise to infinite hazard at the origin nor hazard increasing without bound" (9:306). Kaminsky and Nelson illustrate the use of the Pareto in developing replacement policy. The Pareto can be used to predict component replacement times based on an accumulation of early failure data (17:145).

PARETO FUNCTION

The mathematical formulation of Pareto's Law on income distribution is shown in Eq (3.1). This law corresponds to the following Pareto probability density function as given by Johnson and Kotz:

$$P(x) = Pr[X \geq x] = (a/x)^c \qquad a>0,\ c>0,\ x \geq a \qquad (3.2)$$

In this equation P(x) gives the probability that income is equal to or greater than x, while a corresponds to some minimum income (16:234). The cumulative distribution function (cdf) of X resulting from Eq (3.2) gives the following Pareto distribution:

$$F_X(x) = 1 - (a/x)^c \qquad a>0,\ c>0,\ x \geq a \qquad (3.3)$$

(16:234). During Mandelbrot's investigation concerning the Pareto distribution, he distinguishes between two forms of the Pareto Law: the Strong Law of Pareto and the Weak or Asymptotic form of the Law of Pareto. Mandelbrot's Strong Law of Pareto is of the form shown in Eq (3.3) and is written as follows:

$$\begin{aligned} 1 - F_X(x) &= (x/a)^{-c} && x \geq a \\ &= 1 && x < a \end{aligned} \qquad (3.4)$$

Mandelbrot's Weak or Asymptotic form of the Pareto Law is written as follows:

$$1 - F_X(x) \sim (x/a)^{-c} \quad \text{as} \quad x \rightarrow \infty \qquad (3.5)$$

The Weak form implies that if the log of the left side of the relation is graphed against log x "the resulting curve should be asymptotic to a straight line with slope equal to [-c] as x approaches infinity" (16:245).

Grouping Pareto Distributions by Kind. There are several versions of the Pareto cumulative distribution function. Often, these versions are grouped according to 'kind'. There are three labels used in this type of grouping scheme: Pareto distributions of the first kind, of the

second kind, and of the third kind.

A distribution of the form shown in Eq (3.3) is referred to as a Pareto distribution of the first kind (16:234). A Pareto distribution of the second kind is written as follows:

$$F(x) = 1 - K/[(x + C)^{C}] \tag{3.6}$$

(16:234). This form differs from the Pareto distribution of the first kind through the addition of another quantity, C, in the denominator of the second term on the right hand side of the equation.

In addition to the two distribution kinds above, Pareto suggested a third law, the distribution of which Mandelbrot calls a Pareto distribution of the third kind. The mathematical form is as follows:

$$F(x) = 1 - [k_2 e^{-hx}/(x + C)^{C}] \tag{3.7}$$

(16:234). The Pareto distribution of the third kind degenerates to that of the second kind when $h = 0$.

Grouping Pareto Distributions by Parameter Number. Perhaps a more understandable method of grouping the various forms of the Pareto distribution function is by grouping them according to the number of parameters the form contains. However, before describing these functions, three basic parameters will be defined.

Hastings and Peacock describe three types of parameters which always have a physical or geometrical meaning. These three parameters are those of location (a), scale (b) and shape (c). This study will use this symbology when using these parameters. The location parameter, a, is "the abscissa of a location point (usually the lower or mid point) of

the range of the variate" (15:20). The scale parameter, b, "determines the scale of measurement of the fractile, x" (15:20). A fractile is a general element within the range of the variate, X (15:5). Finally, the shape parameter, c, "determines the shape (in a sense distinct from location and scale) of the distribution function (and other functions) within a family of shapes associated with a specified type of variate" (15:20). Using the normal distribution as an example, the mean is the location parameter because it specifies a kind of mid point for the distribution. The standard deviation is the scale parameter because it provides a fractile measurment device for the distribution. "The normal distribution does not have a shape parameter" (15:20). With this background on location, scale and shape parameters, we can now proceed with the disussion on grouping Pareto distributions according to the number of parameters contained in the distribution expression.

The most commonly used form of the Pareto distribution is the two parameter form; however, there is a more general form which uses all three basic parameters of location (a), scale (b), and shape (c). This section will present this more general form and show how the simpler forms are derived from it. The three parameter form of the Pareto distribution is written as follows:

$$F(x) = 1 - [1 + (x-a)/b]^{-c} \qquad x \geq a \qquad (3.8)$$

where $b>0$ and $a>0$ (20:218). As stated earlier, the notation of Hastings and Peacock is used in this equation and in those that follow.

The two parameter Pareto distribution is the most common form of the distribution and is derived from Eq (3.8) by eliminating either the

location or the scale parameter from the equation. One way to obtain a two parameter distribution function is to set the location parameter equal to zero. For $a=0$ we obtain a Pareto distribution of the second kind as shown in Eq (3.6) where $K=b^C$ and $C=b$. This special case is sometimes referred to as the Lomax distribution (20:218). Another method of effectively eliminating one of the parameters is to set the location parameter equal to the scale parameter. Setting $a=b$ in Eq (3.8) results in the usual formulation of the Pareto distribution and is the Pareto distribution of the first kind as shown in Eq (3.3).

The simplest form of the Pareto distribution is the one parameter version which can be obtained by setting both the location and the scale parameter equal to one. Setting $a=b=1$ in Eq (3.8), the following ditribution function results:

$$F(x) = 1 - x^{-c} \qquad x \geq 1 \qquad (3.9)$$

This one parameter form is regarded as the 'standard form' of the Pareto distribution (16:240).

Since most of the many versions of the Pareto distribution can be derived from the more general three parameter model, this thesis investigates the three parameter distribution. This should ensure that results of this study can be used in a wider variety of applications where estimation is required.

PARAMETER ESTIMATION

This section describes the estimation methods used in this study as applied specifically to the Pareto distribution. First the best linear unbiased estimators are presented along with the procedure used to

transform these estimators into a computational form. Then the minimum distance estimation formulas will be adapted to the Pareto distribution.

Best Linear Unbiased Estimator. As was mentioned by Kulldorff and Vännman, the general, three parameter form of the Pareto cumulative distribution function has received little attention from statisticians working on the development of estimators (20:218). Hence, many estimators have been developed for special cases of the two parameter formulation while few estimators are available for study of the more general distribution form.

Kulldorff and Vännman successfully derived BLU estimators for three cases of the general Pareto distribution where the shape parameter is always assumed to be greater then two. Specifically, these cases are: scale parameter when the location and shape are known; location parameter when the scale and shape are known; and location and scale parameters when the shape is known (20:218-224). The estimators developed for the third case are the estimators used in this study, since only shape parameters will be explicitly specified for the Pareto distribution being investigated. However, these estimators are useful only when $c>2$.

Vännman later presented the BLU estimators for the same three cases shown above with the condition that the shape parameter is equal to or less than two (36:704). Therefore, his estimators will be used for the cases when $c<2$.

BLUEs for Shape Greater Than 2. As stated earlier, Kulldorff and Vännman developed BLU estimators for both the location and scale parameters with the shape parameter known and greater than two. From Chapter II. we recall that the BLU estimator is based on

order statistics where the random variables are arranged in order of magnitude from smallest to largest (24:229). Therefore, the elements of the drawn sample are ordered from smallest to largest to provide the order statistics where $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$. Here $x_{(1)}$ is the smallest valued observation and $x_{(n)}$ is the largest valued observation from the sample of size n. Since a BLU estimator must take the form of a linear combination of the ordered random variables, the BLU estimators for the location and scale parameters of the Pareto distribution with specified shape must be a linear combination of the ordered sample observations, where the coefficients of these observations are to be determined. In developing their BLU estimators, Kulldorff and Vännman derived this linear relationship and determined the coefficients which are based on the sample size and the specified shape parameter. The BLU estimators for location, a, and scale, b, are written as follows:

$$\hat{a} = x_{(1)} - Y/[\,(nc-1)(nc-2) - ncD\,] \qquad (3.10)$$

$$\hat{b} = Y(nc-1) / [\,(nc-1)(nc-2) - ncD\,]$$

$$= (nc-1)[x_{(1)} - \hat{a}] \qquad (3.11)$$

The authors note that in the special case when $a=b$, the BLU estimator reduces to the following:

$$\hat{a} = [\,1 - (1/nc)\,]\, x_{(1)} \qquad (3.12)$$

Equations (3.10) and (3.11) both contain two quantities, Y and D, which still need to be defined. Y is defined in terms of D and an

additional new term B_1, while D simply contains the new term B_1. B_1 is defined in terms of the sample size, n, and the specified shape parameter, c. Therefore, by computing the B_1 terms, both D and Y can be determined. With D and Y known, one can then calculate the BLU estimators of location and scale. The expressions for Y, D and B_1 are as follows:

$$Y = (c+1) \sum_{i=1}^{n-1} B_i x_{(i)} + (c-1) B_n x_{(n)} - D x_{(1)} \qquad (3.13)$$

$$D = (c+1) \sum_{i=1}^{n-1} B_i + (c-1) B_{(n)} \qquad (3.14)$$

$$B_i = \frac{\Gamma(n-i+1)\,\Gamma(n+1-2/c)}{\Gamma(n-i+1-2/c)\,\Gamma(n+1)} \qquad i = 1,2, \ldots ,n \qquad (3.15)$$

Equations (3.10) to (3.15) are the Kulldorff and Vännman equations (20:219-225).

To obtain the BLU estimators, we must calculate all of the B_i values for $i = 1, 2, \ldots, n$. Eq (3.15) shows that B_i contains four gamma functions which would require considerable computational time; however, the expression can be simplified to reduce the computational load.

Banks and Carson indicate that "the gamma function can be thought of as a generalization of the factorial notion which applies to all positive numbers, not just integers" (1:144). They show that for any positive real number, p, the gamma function of p is as follows:

$$\Gamma(p) = (p-1)\,\Gamma(p-1) \qquad (3.16)$$

Since $\Gamma(1) = 1$, we see that if p is an integer than Eq (3.16) reduces to (1:144):

$$\Gamma(p) = (p-1)! \qquad (3.17)$$

Eq (3.16) and Eq (3.17) will be used to simplify the B_i term to a more manageable computational form. The first gamma function in the numerator and the last gamma function in the denominator will always be gamma functions of integer values; therefore, Eq (3.17) can be used to transform these terms to common factorial terms. Eq (3.16) will be used on the remaining gamma function in the numerator to assist in the reduction process. Simplification of the first two B_i terms (30) will reveal a pattern which will simplify the evaluation process:

$$B_1 = \frac{\Gamma(n-1+1)\,\Gamma(n+1-2/c)}{\Gamma(n-1+1-2/c)\,\Gamma(n+1)}$$

$$= \frac{\Gamma(n)\,\Gamma(n+1-2/c)}{\Gamma(n-2/c)\,\Gamma(n+1)}$$

$$= \frac{(n-1)!\;\;(n-2/c)\;\;\Gamma(n-2/c)}{n\,(n-1)!\;\Gamma(n-2/c)}$$

$$= (n - 2/c) / n \;\; = \;\; 1 - 2/(cn) \qquad (3.18)$$

Solving for B_2 in a similar manner yields the following:

$$B_2 = \frac{\Gamma(n-2+1)\,\Gamma(n+1-2/c)}{\Gamma(n-2+1-2/c)\,\Gamma(n+1)}$$

$$= \frac{\Gamma(n-1)\,\Gamma(n+1-2/c)}{\Gamma(n-1-2/c)\,\Gamma(n+1)}$$

$$= \frac{(n-2)!\;\;(n-2/c)\;\;\Gamma(n-2/c)}{n!\;\Gamma(n-1-2/c)}$$

$$= \frac{(n-2)!\;\;(n-2/c)\;(n-1-2/c)\;\;\Gamma(n-1-2/c)}{n\;(n-1)\;(n-2)!\;\;\Gamma(n-1-2/c)}$$

$$= \frac{(n\;-\;2/c)\;(n\;-\;1\;-\;2/c)}{n\;(n\;-\;1)}$$

$$= [\;1\;-\;2/(cn)\;]\;\;[\;1\;-\;2/c(n-1)\;] \qquad (3.19)$$

Equations (3.18) and (3.19) reveal the following pattern for the B_n.

$$B_n = [1 - 2/cn]\,[1 - 2/c(n-1)]\;.\;.\;.\;[1 - 2/c(1)] \qquad (3.20)$$

The following notation will allow even further simplification of the B value computations:

Let $t_1 = 2/c(n)$, $t_2 = 2/c(n-1)$, . . . , $t_n = 2/c(1)$.

Let $u_1 = 1 - t_1$, $u_2 = 1 - t_2$, . . . , $u_n = 1 - t_n$.

Then $B_1 = u_1$, $B_2 = u_1u_2$, . . . , $B_n = u_1u_2 \ldots u_n$.

And in general, the computational form is as follows:

$$B_i = \prod_{j=1}^{i} u_j \qquad (3.21)$$

where $u_j = 1 - t_j$ and $t_j = 2/c(n-j+1)$ for $j = 1, 2, \ldots, i$ (30). Equipped with these relations, we can now write the following recursive relationship which will allow simpler calculations as recommended by Vännman (36:705):

$$B_i = [1 - 2/c(n-i+1)]\, B_{i-1} \qquad i = 1,2,\ldots,n \qquad (3.22)$$

With these relationships available, the programming of these calculations will be much simpler.

BLUEs for Shape Equal to or Less Than 2. Vännman indicates that the variance of the Pareto distribution does not exist when the shape parameter, c, is equal to or less than 2; therefore, the above formulas for BLU estimators of location and scale do not apply. He further states, however, that if only the first k order statistics are used in the estimator, where $2 \leq k < (n+1 - 2/c)$, then the variance of the estimator does exist, with the added condition that the shape parameter satisfies the following relationship: $2/n < c \leq 2$. He indicates that the most efficient estimator is obtained by basing the estimator on the first k order statistics where $k = n - [2/c]$. In this equation the bracketed fraction implies that only the integer

portion of the fraction is used in the calculation (36:705-707). The formulas for the location and scale parameters based on the first k order statistics are as follows (36:707):

$$a_k = x_{(1)} - b_k/(nc-1) \tag{3.23}$$

and

$$b_k = (1/U_k) \{ (c+1) \sum_{i=1}^{k-1} B_i x_{(i)} + [(n-k+1)c - 1] B_k x_{(k)} - [(nc-1)/(nc)] (nc - 2 - U_k) x_{(1)} \} \tag{3.24}$$

where

$$U_k = \frac{(nc-2)(nc-c-2) - nc[(n-k)c - 2]B_k}{(nc-1)(c+2)} \tag{3.25}$$

Again, Equations (3.23) and (3.24) can only be used where k represents the first k order statistics and where $k < n + 1 - 2/c$. To obtain the most efficient BLU estimator, Vännman indicates that k should additionally staisfy the following: $k = n - [2/c]$. He further states that in the case where 2/c is already an integer value, then eq (3.24) simplifies to the following (36:707):

$$b = \frac{(c+1)(c+2)(nc-1)}{(nc-2)(nc-c-2)} \{ \sum_{i=1}^{n-2/c} B_i x_{(i)} - [(nc-2)/(c+2)] x_{(1)} \} \tag{3.26}$$

Eq (3.26) can then be entered into Eq (3.23) to obtain the BLU estimator for location. However, to use eq (3.26), the simplified version of the BLU estimator of the scale parameter, four conditions must exist:

1) The shape parameter, c, must be specified

2) $2/n < c \leq 2$

3) $2/c$ is an integer

4) $2 \leq k = n - 2/c$

Finally, Vännman notes two simplified expressions for B_i when the shape parameter equals 1 or 2. He indicates that if $c = 1$, then $B_i = (1 - i/n)[1 - (i-1)/n]$. If $c = 2$, then $B_i = 1 - i/n$ (36:705). The computer program verification and validation phase of this research revealed that Vännman's simplified expression for B when $c = 1$ was incorrect in the published reference. By setting $c = 1$ and simplifying Eq (3.15), the error in Vännman's published formula was found. To generate correct B values with $c = 1$, Vännman's bracketed term $[1 - (i-1)/n]$ must be changed to $[1 - i/(n-1)]$. These simplified expressions will be valuable in the computer programming phase of this study, since B values must be calculated to determine the BLU estimates.

Minimum Distance Estimator. The general computational forms of the three distance measures used in this study are presented in Chapter II and are reflected in equations (2.7), (2.9) and (2.10). To apply these measures using a Pareto distribution, we simply substitute our hypothesized Pareto distribution function, P_i, for the z_i value currently shown in these equations, where the starting point for the estimates of location and scale will be the BLU estimates. This hypothesized Pareto cdf can be written as follows:

$$P_i = F(x_{(i)};\hat{a},\hat{b},c) = 1 - [1 + (x_{(i)} - \hat{a})/\hat{b}]^{-c} \qquad (3.27)$$

The minimization routine, ZXMIN, from the the International Mathematical Statistics Library (IMSL) will then alter the values of location and scale to obtain the minimum distance measure values. These altered estimates for location and scale then become the minimum distance estimates for that particular distance measure. The procedural details are covered in more depth in the following chapter.

IV. Monte Carlo Analysis

This chapter will describe the specific analysis tool used in this study to compare the best linear unbiased and the minimum distance estimation techniques. The tool is called Monte Carlo analysis. Following a general discussion of the Monte Carlo method, the specific application of the method in this study will be described. This application description will present the three step process of Monte Carlo analysis along with the detailed procedures involved within each step.

MONTE CARLO METHOD

The Monte Carlo method, or the method of statistical trials (6:1), falls within the realm of experimental mathematics. Hammersley and Handscomb indicate that the essential difference between theoretical and experimental mathematicians "is that theoreticians deduce conclusions from postulates, whereas experimentalists infer conclusions from observations" (13:1). Monte Carlo analysis is a member of the experimental mathematics branch since it deals with mathematical experiments on random numbers (13:2). A further explanation of the Monte Carlo method is provided by Schreider:

> The Monte Carlo method (or the method of statistical trials) consists of solving various problems of computational mathematics by means of construction of some random process for each such problem, with the parameters of the process equal to the required quantities of the problem. These quantities are then determined approximately by means of observations of the random process and the computation of its statistical characteristics, which are approximately equal to the required parameters [6:1].

This description of the Monte Carlo method reflects how well suited the method is for this particular study, since the description mirrors the process used to compare the two estimation techniques.

MONTE CARLO STEPS AND PROCEDURES

This study uses a three step Monte Carlo process to compare best linear unbiased estimation with minimum distance estimation (using three distinct distance measures) as applied to the Pareto distribution. First, one generates random variates from a specified Pareto distribution (i.e., a Pareto distribution with known parameters). Second, the two estimation techniques are used to obtain parameter estimates based on the random sample data from the first step. Third, the resulting estimates are compared to determine which estimation technique provided the better parameter estimates (4:27).

Step 1: Data Generation. Using the Monte Carlo technique, we generate our own random data using the random number generator of the VAX 11/785 (VMS) computer system located at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. A random number generator generates random numbers uniformly distributed on [0,1] (1:293). Parr stated that there were four items required to perform a minimum distance estimation: a set of data, a parametric model, a distance measure, and a minimization routine (26:1207-1208). The data generation step supplies the first two items by generating the data based on a specified parametric model, the Pareto distribution.

In the first step, the researcher generates the random sample data needed to create the controlled environment, using different parameter values for each data set. To evaluate the effect of sample size on the

estimators and ensure validity, sample sizes (n) of 6, 9, 12, 15, and 18 are used. Additionally, shape parameters (c) of 1.0, 2.0, 3.0, and 4.0 are used with the location parameter (a) set to 1 and the scale parameter (b) set to 1 for each sample size resulting in 20 total data sets. The random sample data required for the study are random variates from a specified Pareto distribution. Previous thesis students had used distributions for which computer programs were already available to generate random variates using subroutines from the International Mathematical Statistics Library (IMSL) (4:27; 18:43). However, IMSL does not contain a similar subroutine for the Pareto distribution. Therefore, the random variate relationship was derived using the inverse transform technique (1:294-295) on the general three parameter Pareto distribution function shown in Eq (3.8) with location parameter of 1 and scale parameter of 1. The derivation of the Pareto random variate relationship begins by substituting a=1 and b=1 into Eq (3.8) which yields the following:

$$F(x) = 1 - (1/x)^c \tag{4.1}$$

Letting R be a random number between 0 and 1 and letting X be the random variate, we have:

$$R = 1 - (1/X)^c \tag{4.2}$$

Solving for X yields the Pareto random variate relationship:

$$X = (1/R)^{1/c} \tag{4.3}$$

For each of the 20 data sets, 1000 samples are generated where each

data set is characterized by a unique sample size (n = 6, 9, 12, 15, or 18) and shape parameter (c = 1.0, 2.0, 3.0, or 4.0) with location parameter and scale parameter set equal to 1. Therefore, a total of 20000 random sample sets are generated, since 20 separate data sets are required to reflect all the combinations of sample sizes and shape parameters. Previous studies also used 1000 samples to evaluate the estimation techniques (4:28; 18:43). A computer subroutine, PARVAR, was written to generate the 20000 random sample sets from the three parameter Pareto distribution. The IMSL subroutine VSRTA was used on each sample set of size n to arrange the random variates from smallest to largest. The output was then used by each of the estimation technique subroutines.

Step 2: Estimate Computation. The second step of the Monte Carlo process is to use both of the estimation techniques, best linear unbiased and minimum distance estimation, to compute estimates based on the random sample data sets. We first present the procedures used for finding the best linear unbiased estimates. This presentation is followed by the minimum distance estimation procedures.

Using each of the data sets along with the best linear unbiased estimators for the location and scale parameters of the Pareto distribution function for each data set, one obtains 1000 best linear unbiased estimates for the parameters of each particular Pareto distribution sampled. The computer subroutines written to perform this task were titled BLCGT2 and BLCLE2. These subroutines were eventually run against all 20 data groups.

The minimum distance estimation process develops six minimum distance estimators using the 'BLUE' estimates of location and scale

for each sample of size n as the starting values for the hypothesized distribution function, $F(x_1;\hat{a},\hat{b},c)$, which in our computational notation is equal to z_1. The IMSL minimization subroutine, ZXMIN, then minimizes the computational form of each distance measure in turn. For instance, by varying the value of the location parameter while holding the scale equal to the BLUE for scale, ZXMIN finds the value of the location parameter which minimizes the distance between the hypothesized distribution and the empirical distribution function for each sample of size n. This new value for the location parameter is the single parameter minimum distance estimate of the location parameter. Alternatively, by holding the location parameter equal to the BLUE for location, ZXMIN uses the same procedures to obtain a single parameter minimum distance estimate for the scale parameter. Finally, ZXMIN finds what we call a double parameter minimum distance estimate by varying both the location and the scale parameters in the same minimization calculation. The result of a double parameter minimum distance estimate run is a simultaneous estimate of both location and scale. The two single parameter minimization techniques (i.e. one for location and one for scale) along with the double parameter minimization technique are applied to each of the three distance measures, resulting in 12 minimum distance estimates for each data set generated. The computer subroutines written to perform these tasks are KSMD, KSAMD, and KSBMD for the Kolmogorov distance measure. For the Cramer-von Mises distance measure, the subroutines CVMD, CVAMD, and CVBMD were written. Finally, for the Anderson-Darling distance measure, the subroutines written are entitled ADMD, ADBMD, ADABMD. The source code for these subroutines is located in Appendix B. Each of these subroutines is run against all 20

data groups.

Step 3: Estimate Comparison. The third and final step in the Monte Carlo analysis is estimate comparison. In this step, the mean square error (MSE) approach is used to evaluate which estimation technique provides more accurate parameter estimates (4:31).

Many statisticians support the use of MSE as a good evaluation for comparing estimators. Mendenhall and Scheaffer state that MSE is the expected value of $(\hat{\theta} - \theta)^2$. They further indicate that the mean square error can be written as the summation of the variance and the square of the bias of an estimator (24:267). Since we seek unbiased and relatively efficient estimators, small MSE values should provide a good indication of estimators possessing these two desirable properties and should therefore provide a good estimator comparison tool. Mendenhall further describes a method for evaluating an estimator which parallels the method used in this study:

> Thus the goodness of a particular estimator could be evaluated by testing it by repeatedly sampling a very large number of times from a population where the parameters were known and obtaining a distribution of estimates about the true value of the parameter. This distribution of estimates would be referred to as the 'distribution of the estimator' . . . Those estimators possessing distributions that grouped most closely about the parameter would be regarded as 'best' . . . Hence, the relative 'goodness' of estimators may be evaluated by comparing their biases and their variances [23:14-15].

Since the MSE is a function of both the variance and the bias, an MSE comparison should reflect the goodness of the estimators considered, as suggested by Mendenhall. However, Mood and Graybill warn that "except in trivial cases, there is no estimator whose mean-squared error is a minimum for all values of θ" (25:167). That is, for a given θ value, estimator A may produce the smallest MSE, while for another value of the parameter, estimator B may provide the smallest MSE. However, Mood

and Graybill do concede that the MSE does provide a useful guide for estimator comparisons. In fact, they do end up using the MSE as their guide in searching for minimum risk estimators. Finally, Liebelt provides two reasons why minimizing the average mean square error is a credible criteria for evaluating good estimators:

> First, if the mean square error is zero or near zero then the dispersion of the estimate from the true value is also zero or near zero. Secondly, the choice of minimizing the average mean square error is an easy mathematical procedure, whereas other choices often lead to insurmountable analytical difficulty [21:137].

Therefore, many authors support the use of comparative mean square errors as a valid technique for evaluating the relative worth of estimators, where the estimator with the smallest MSE is considered the 'best' estimator for a given set of parameter values.

The term mean square error is very descriptive of the procedures used during the evaluation. The 'error' from each of the 1000 samples of size n is found by subtracting the estimated parameter from the true population parameter. This error term is then squared, giving the 'square error.' Finally, the mean of the 1000 'square error' terms is found by summing these terms and dividing by 1000, thereby producing a 'mean square error' (4:32). The estimator providing the smallest MSE, therefore, is the best estimation technique to use. The formula for calculating the MSE is as follows:

$$MSE(\hat{\theta}) = [\sum_{i=1}^{N}(\hat{\theta}_i - \theta)^2]/N \qquad (4.4)$$

"where θ is the true value of the parameter, $\hat{\theta}_i$ is the ith estimate, and N is the number of times the estimation is performed--in this

analysis, N = 1000 " (4:32). In this case, the parameters being evaluated are the location and scale parameters. Of course, the computer was used to perform the MSE calculations because of the large number of calculations involved. The MSE calculations are embedded in the main program, BLUMD, thereby eliminating the need to store large numbers of variate and estimate values. The MSE calculations result in seven estimation error indicators for the location parameter and seven estimation error indicators for the scale parameter for every specified Pareto distribution considered. The seven estimation error indicators for each parameter correspond to the seven estimation techniques used: the best linear unbiased estimator and the six minimum distance estimators. The estimation technique which reflects the smallest MSE is considered the best parameter estimation technique for that specified Pareto distribution.

The subroutines described above in the three step Monte Carlo analysis were merged into a computer program (BLUMD) which output the MSE values for the estimation techniques being compared. The source code is found in Appendix B. The logical steps or pseudocode for the program is listed in Figure 1.

Each of the subroutines in BLUMD were validated and verified individually by comparison with sample hand calculations. Additionally, the subroutines were again validated and verified as they were added to the parent program. It was this validation and verification procedure which first indicated there were possible problems with Vannman's published B value formula which supported the generation of the best linear unbiased estimates for c=1. Chapter 3 identifies the published version of the B value formula and the correction required.

Steps in BLU vs MD Estimate Comparison

1. Generate a sample set of n random variates from a Pareto distribution with location and scale equal to 1 and shape equal to c.

2. Order sample from smallest to largest.

3. Calculate BLU estimates for location and scale based on sample size n.

4. Calculate the Kolmogorov minimum distance estimates of location and scale based on the sample.

5. Calculate the Cramer-von Mises minimum distance estimates of location and scale based on the sample.

6. Calculate the Anderson-Darling minimum distance estimates of location and scale based on the sample.

7. Find the error from the true value of 1 for each estimate and square this error. Save a running sum of the squared error terms for each estimate.

8. Repeat steps 1-7 1000 times for a given n.

9. Divide all eight squared error totals by 1000 to give the MSE's.

10. Output the 14 MSE's for the given n and c values.

11. Repeat steps 1-10 using a different sample size, n, but the same c, until all values of n have been used.

12. Repeat steps 1-11 using a new shape value, c, until all values of c have been used.

Figure 1. Pseudocode for Program BLUMD

V. Results, Analysis and Conclusions

Figure 1 in Chapter IV. described the pseudocode of the computer program used to generate each Pareto random variate sample set, calculate the best linear unbiased and the six new minimum distance estimates for each parameter based on each sample set, and finally determine the mean square error for each estimate. This chapter presents the results of the computer program runs along with an analysis of these results. Appendix A contains the results of the study in table format, where a separate table of Mean Square Error values is presented for each unique shape parameter--sample size combination investigated. Since there were 20 possible combinations of shapes and sample sizes, Appendix A contains 20 separate tables. Finally, this chapter presents the conclusions drawn from the analysis of these results.

RESULTS

Appendix A contains the tables of mean square errors (MSEs) for each estimation technique used in the research effort, given a particular shape parameter and sample size. Since MSE is the evaluation tool used to determine which estimation technique was best, these tables were used to make the estimator comparisons. The estimator with the smallest MSE value is considered the best estimator of those investigated.

TABLE II

Mean square error for c = 1 and n = 9

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	2.1462625E-02	BLUE	0.4062696
ADMD1	2.3038462E-02	CVMD1	0.5638790
CVMD1	2.4678135E-02	ADMD1	0.5760345
KSMD1	4.0461082E-02	KSMD1	0.5824831
CVMD2	4.2301375E-02	KSMD2	0.6407889
KSMD2	4.3836664E-02	CVMD2	0.6685479
ADMD2	6.1653644E-02	ADMD2	0.8271174

Figure 2. Sample Table of Mean Square Errors

Figure 2 shows a sample of the table format. Each table contains two sections. The left section contains the MSEs based upon estimation of the location parameter while the right section contains the MSEs based upon estimation of the scale parameter. This format permits an easy comparison of the BLUE MSE value for each parameter with the MSE value of each of the minimum distance estimation techniques used. The smallest MSE value in each table section then reflects the best estimation technique to use for that parameter, under the stated shape parameter and sample size conditions. To further simplify the reading of the tables, the estimation techniques are ordered in each table from smallest MSE value to largest. Therefore, the technique which generated the smallest MSE result is listed first in each column and is also the

best estimation technique to use for estimating that parameter under the specified conditions.

Each section of the table contains a list of the estimation techniques that were applied to the 1000 sample sets of ordered Pareto random variates. BLUE refers to the best linear unbiased estimation technique. Each of the other techniques was compared against this technique to determine which technique provided the better estimate. ADMD1 refers to the Anderson-Darling minimum distance estimation technique. Additionally, the 1 implies that only one parameter was permitted to vary while the other parameter was held constant (equal to the BLU estimate). For example, ADMD1 under the location parameter section of the table implies that the location parameter was varied while the scale parameter was held equal to the BLU estimate for that sample size and shape parameter. CVMD1 and KSMD1 refer to the Cramer-von Mises and the Kolmogorov minimum distance estimation techniques respectively. Again, the 1 implies that only one parameter was allowed to vary in finding the minimum distance value, while the other parameter was held equal to the BLU estimate. ADMD2 again refers to the use of the Anderson-Darling distance measure in the minimization process. However, in this case, both the location and scale parameters were permitted to vary simultaneously in determining the minimum distance measure. The 2 in the notation indicates that two parameters were allowed to vary during the minimization process. CVMD2 and KSMD2 refer to the use of the Cramer-von Mises and Kolmogorov distance measures respectively. Again, the 2 in the notation implies that two parameters (i.e., location and scale) were permitted to vary during the

minimization routine. Therefore, in this report, ADMD1, CVMD1 and KSMD1 are called single parameter minimum distance estimation techniques while ADMD2, CVMD2 and KSMD2 are called double parameter minimum distance estimation techniques.

ANALYSIS

Regarding the location parameter, the BLU estimator provided the smallest MSE values in all cases except the case where the shape parameter equalled 1 (c = 1) and the sample size equalled 6 (n = 6). In this case, the single parameter Anderson-Darling minimum distance estimator (ADMD1) provided the smallest MSE. Based on this analysis, the results showed that, overall, the best linear unbiased estimator performed better than any of the minimum distance estimators evaluated.

The results of the research regarding estimation of the scale parameter was even more pronounced. Regardless of the shape parameter (c = 1, 2, 3 or 4) or sample size (n = 6, 9, 12, 15, or 18) used in this study, the BLUE provided the smallest MSE in every case and is therefore ranked as the best of the estimation techniques investigated. None of the minimum distance estimation techniques provided better MSE values in any instance. Therefore, investigators should feel comfortable using the BLUE as an instrument of estimation when the underlying population distribution is the Pareto.

Additionally, some observations were made regarding the minimum distance estimation techniques that were applied in this study and how they performed against each other. Performance of the minimum distance estimators on both location and scale were addressed.

For the location parameter, the single parameter Anderson-Darling minimum distance estimator (ADMD1) provided the smallest MSE values in every case among the minimum distance estimators tested. ADMD1 was therefore considered the best minimum distance estimator of location among those investigated.

One concern this researcher had regarding the minimum distance estimation technique was whether to let both the location and scale parameters vary (double parameter estimator) to achieve the minimum distance measure or to permit only one of the parameters to vary (single parameter estimator) while holding the other as a constant, equal to the BLUE for that parameter. For the location parameter, the results show that the single parameter minimum distance estimator out-performed its double parameter counterpart in every case except one. When $c=1$ and $n=6$, KSMD2 provided a smaller MSE than did KSMD1. In all other cases, however, the single parameter minimum distance estimator provided better results. Therefore, the single parameter estimation technique performed better than its double parameter counterpart when the Kolmogorov, Cramer-von Mises, or Anderson-Darling distance measure was minimized for location parameter estimation of the Pareto.

For the scale parameter, the inferences drawn required a bit more scrutiny as there was no single best minimum distance estimator. There was a shift in performance when a shape parameter $c > 1$ was specified. For $c = 1$, CVMD1 was the best overall estimator, since it provided the smallest MSE in four of the five cases investigated. The exception occurred again in the case $c=1$ and $n=6$, where the KSMD1 estimator gave the smallest MSE; however, CVMD1 did provide the next smallest MSE.

Therefore, CVMD1 was selected as the best minimum distance estimator for scale when the shape was specified as $c = 1$. In the other 15 cases investigated, KSMD2 provided the smallest MSE values in 12 instances. The three exceptions were: $c=2$ and $n=6$ where KSMD1 was best, $c=2$ and $n=15$ where CVMD1 was best, and $c=3$ and $n=15$ where ADMD1 was best. Overall, CVMD1 performed best for $c=1$ and KSMD2 performed best for $c=2$, 3 or 4 among the minimum distance estimators for scale investigated.

Regarding use of the single parameter versus the double parameter minimum distance estimation technique for scale, no clear rule can be stated, although there was a definite trend shown in the results. For $c=1$, the single parameter technique clearly dominated since in all but one case, the single parameter estimator provided smaller MSE values than the corresponding double parameter estimator. The only exception was for $c=1$ and $n=12$ where KSMD2 performed better than KSMD1 (perhaps an indication of the improved performance this estimator would show for larger c values). However, as the shape parameter value increased from 1 to 4, the performances of the double parameter techniques improved. In fact for $c=4$, the double parameter estimation techniques performed better than their single parameter counterparts in all but one case: for $c=4$ and $n=18$, ADMD1 out-performed ADMD2. Therefore, for $c=1$, the single parameter minimum distance estimators performed better overall than their double parameter counterparts. For $c=4$, the reverse was true. For shape values of 2 and 3, the performance was mixed, but the trend toward improved double parame.er performance with the increasing value of shape was still evident.

CONCLUSION

The observations made regarding the minimum distance estimators, although lengthy, should not overshadow the primary conclusion drawn from this research. The best linear unbiased estimators provided the best estimates of both location and scale when compared with any of the minimum distance estimators based upon the mean square error criteria.

VI. Summary and Recommendations

This chapter presents a summary of the research effort, restating the objective of the study, the methodology used, and the major conclusions drawn from the experimental results. Further, three recommendations for further study in this area are presented.

SUMMARY

The purpose of this research was to compare the minimum distance estimation technique with the best linear unbiased estimation technique to determine which estimator provided more accurate estimates of the underlying location and scale parameter values for a given three parameter Pareto distribution with specified shape parameter. The Kolmogorov, Cramer-von Mises, and Anderson-Darling distance measures were used to develop the minimum distance estimators. For each of these distance measures, two minimum distance estimators were developed. The first minimum distance estimator varied only a single parameter value to achieve the distance measure minimization. This estimator was called the single parameter minimum distance estimator. The second minimum distance estimator allowed both the location and scale parameters to vary while achieving the minimum distance measure. These estimators were called the double parameter minimum distance estimators. These minimum distance estimators were compared against the best linear unbiased estimators which had been previously developed by Kulldorff and Vannman for shape greater than 2, and by Vännman for shape equal to or less than 2. Manual derivation of the B values formula for the special

case, c=1 , revealed an error in the published version of the formula, as explained in Chapter III. of this report.

A Monte Carlo methodology was used to generate the estimates for each of the estimation techniques investigated. A sample of Pareto random variates was generated from a completely specified three parameter Pareto distribution with location and scale equal to one and the shape parameter iteratively specified as one, two, three, or four. The estimates of location and scale were then generated based on each of the estimation techniques. This process was repeated 1000 times for each combination of shape parameter (c = 1, 2, 3, or 4) and Pareto random variate sample size (n = 6, 9, 12, 15, or 18). This Monte Carlo process resulted in 1000 estimates of both location and scale for each estimation technique used.

The criteria for determining which estimation technique performed best was based on the resulting mean square error calculation for each group of 1000 estimates. The estimation technique which yielded the smallest mean square error was selected as the best performing estimator.

The results of this research clearly indicated that the best linear unbiased estimator provided smaller mean square error terms than any of the minimum distance estimation techniques investigated. Therefore, the best linear unbiased estimation technique was ranked as the best estimation technique among those tested.

Regarding the minimum distance estimators, a comparison of the single versus double parameter techniques was made. For estimation of the location parameter, the single parameter estimation technique

performed better than the double parameter estimation technique. For the scale parameter estimation, the conclusion was not as clear. A trend was identified as the value of the specified shape parameter increased from 1 to 4. For $c = 1$, the single parameter estimators performed better; however, as the shape parameter increased, the performance of the double parameter estimators improved until at $c = 4$, the double parameter estimators performed better than their single parameter counterparts.

RECOMMENDATIONS

Three recommended areas for further study in this research area are now offered. First, a study similar to this one can be performed, again based upon a specified three parameter Pareto distribution, but using minimum distance estimators based on different distance measures. Examples of such distance measures include the Kuiper distance and the Watson distance referenced by M. A. Stephens (35:731). Second, a study involving a comparison of a set of minimum distance estimators against the best linear unbiased estimators based on the more commonly used two parameter form of the Pareto distribution could prove fruitful. Third, a researcher could perform a comparison study involving the maximum likelihood estimator and a set of minimum distance estimators, again based upon the common two parameter form of the Pareto distribution function. Any of these areas would provide fertile ground for the investigative statistical researcher.

Appendix A

Tables of Mean Square Errors

The following notation is used in this appendix:

Term	Notation
Best Linear Unbiased Estimator	BLUE
Anderson-Darling Minimum Distance Estimator (Only one varying parameter)	ADMD1
Cramer-von Mises Minimum Distance Estimator (Only one varying parameter)	CVMD1
Kolmogorov Minimum Distance Estimator (Only one varying parameter)	KSMD1
Anderson-Darling Minimum Distance Estimator (Two varying parameters)	ADMD2
Cramer-von Mises Minimum Distance Estimator (Two varying parameters)	CVMD2
Kolmogorov Minimum Distance Estimator (Two varying parameters)	KSMD2
Location Parameter	a
Scale Parameter	b
Shape Parameter	c
Sample Size	n
Mean Square Error	MSE

The Monte Carlo analysis involves 1000 iterations for the generation of each table. The true value of the location parameter is one and the true value of the scale parameter is one for all of the tables.

TABLE I

Mean square error for c = 1 and n = 6

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
ADMD1	6.1960317E-02	BLUE	0.9352127
CVMD1	6.4573184E-02	KSMD1	1.460597
BLUE	6.7886792E-02	CVMD1	1.523946
CVMD2	9.8166950E-02	ADMD1	1.589061
KSMD2	1.192953 E-01	KSMD2	1.657526
KSMD1	1.2137F4 E-01	CVMD2	1.667902
ADMD2	4.874095 E-01	ADMD2	3.635443

TABLE II

Mean square error for c = 1 and n = 9

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	2.1462625E-02	BLUE	0.4062696
ADMD1	2.3038462E-02	CVMD1	0.5638790
CVMD1	2.4678135E-02	ADMD1	0.5760345
KSMD1	4.0461082E-02	KSMD1	0.5824831
CVMD2	4.2301375E-02	KSMD2	0.6407889
KSMD2	4.3836664E-02	CVMD2	0.6685479
ADMD2	6.1653644E-02	ADMD2	0.8?71174

TABLE III

Mean square error for c = 1 and n = 12

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	1.0628480E-02	BLUE	0.3627600
ADMD1	1.1631799E-02	CVMD1	0.4933714
CVMD1	1.3175875E-02	ADMD1	0.5042603
CVMD2	2.2465685E-02	KSMD2	0.5136845
KSMD1	2.3550959E-02	KSMD1	0.5234504
KSMD2	2.4744846E-02	CVMD2	0.5422385
ADMD2	3.2977652E-02	ADMD2	0.6380025

TABLE IV

Mean square error for c = 1 and n = 15

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	7.4139438E-03	BLUE	0.3053709
ADMD1	9.5694885E-03	CVMD1	0.3754165
CVMD1	1.1153221E-02	ADMD1	0.3911322
KSMD1	1.6373332E-02	KSMD1	0.4156374
KSMD2	2.4338266E-02	CVMD2	0.5204881
CVMD2	2.5020871E-02	KSMD2	0.5390435
ADMD2	2.9120248E-02	ADMD2	0.6280289

TABLE V

Mean square error for c = 1 and n = 18

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	4.3005478E-03	BLUE	0.1737252
ADMD1	5.2362322E-03	CVMD1	0.2037124
CVMD1	6.4189113E-03	KSMD1	0.2041911
ADMD2	8.6196102E-03	ADMD1	0.2043972
KSMD1	1.0911038E-02	KSMD2	0.2185375
CVMD2	1.1255259E-02	CVMD2	0.2299257
KSMD2	1.2906388E-02	ADMD2	0.2362998

TABLE VI

Mean square error for c = 2 and n = 6

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	1.1706045E-02	BLUE	0.5737305
ADMD1	1.2802768E-02	KSMD1	0.7080300
CVMD1	1.3687569E-02	CVMD1	0.7609550
KSMD1	1.5005208E-02	CVMD2	0.7871752
CVMD2	1.7071828E-02	ADMD1	0.8051341
KSMD2	2.2394231E-02	KSMD2	0.9329775
ADMD2	2.8150991E-02	ADMD2	1.161503

TABLE VII

Mean square error for c = 2 and n = 9

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	4.6055736E-03	BLUE	0.2564761
ADMD1	5.1687085E-03	KSMD2	0.2969004
CVMD1	5.9630712E-03	CVMD1	0.3137060
CVMD2	6.9629652E-03	KSMD1	0.3169191
KSMD1	6.9843503E-03	CVMD2	0.3197137
KSMD2	7.8367852E-03	ADMD1	0.3233747
ADMD2	8.9378590E-03	ADMD2	0.4040874

TABLE VIII

Mean square error for c = 2 and n = 12

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	1.9467097E-03	BLUE	0.1996142
ADMD1	2.2874754E-03	KSMD2	0.2196842
CVMD1	2.9733009E-03	ADMD1	0.2351739
ADMD2	3.0276354E-03	CVMD1	0.2386533
KSMD1	3.4914461E-03	KSMD1	0.2393236
CVMD2	3.7764474E-03	CVMD2	0.2409351
KSMD2	4.6899226E-03	ADMD2	0.2438605

TABLE IX

Mean square error for c = 2 and n = 15

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	1.2336310E-03	BLUE	0.1682298
ADMD1	1.4523189E-03	CVMD1	0.1976805
CVMD1	1.9546540E-03	ADMD1	0.1981037
ADMD2	2.0540049E-03	KSMD1	0.2068403
KSMD1	2.3178468E-03	CVMD2	0.2084063
CVMD2	2.6572768E-03	KSMD2	0.2095122
KSMD2	3.2847996E-03	ADMD2	0.2135706

TABLE X

Mean square error for c = 2 and n = 18

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	7.7167596E-04	BLUE	0.1260424
ADMD1	9.3843893E-04	KSMD2	0.1401054
ADMD2	1.3170647E-03	ADMD1	0.1430691
CVMD1	1.3989438E-03	CVMD1	0.1444666
KSMD1	1.6159606E-03	CVMD2	0.1470691
CVMD2	1.8582337E-03	KSMD1	0.1481810
KSMD2	2.3905281E-03	ADMD2	0.1504120

TABLE XI

Mean square error for c = 3 and n = 6

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	4.6162526E-03	BLUE	0.3430938
ADMD1	4.7417325E-03	KSMD2	0.3799514
CVMD1	4.9956846E-03	KSMD1	0.4261144
CVMD2	5.5936114E-03	CVMD2	0.4292807
KSMD1	5.6171883E-03	CVMD1	0.4521263
KSMD2	6.1545176E-03	ADMD1	0.4634998
ADMD2	1.7377743E-02	ADMD2	0.7602041

TABLE XII

Mean square error for c = 3 and n = 9

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	2.5504350E-03	BLUE	0.2173062
ADMD1	2.9328801E-03	KSMD2	0.2343150
CVMD1	3.2298244E-03	CVMD2	0.2613097
KSMD1	3.4083289E-03	CVMD1	0.2633551
CVMD2	3.4843122E-03	KSMD1	0.2647687
ADMD2	3.6204350E-03	ADMD1	0.2666046
KSMD2	3.8682341E-03	ADMD2	0.2911893

TABLE XIII

Mean square error for c = 3 and n = 12

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	1.1032750E-03	BLUE	0.1465466
ADMD1	1.3244717E-03	KSMD2	0.1634654
ADMD2	1.5484567E-03	CVMD2	0.1717137
CVMD1	1.6325671E-03	ADMD1	0.1786888
CVMD2	1.8109155E-03	CVMD1	0.1787402
KSMD1	1.8245766E-03	ADMD2	0.1813732
KSMD2	2.1554409E-03	KSMD1	0.1861933

TABLE XIV

Mean square error for c = 3 and n = 15

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	6.1996793E-04	BLUE	0.1280874
ADMD1	8.2581321E-04	ADMD1	0.1454654
ADMD2	1.0506152E-03	CVMD1	0.1467468
CVMD1	1.0802834E-03	KSMD2	0.1484211
KSMD1	1.2737850E-03	KSMD1	0.1519526
CVMD2	1.2966762E-03	CVMD2	0.1521726
KSMD2	1.3728395E-03	ADMD2	0.1580111

TABLE XV

Mean square error for c = 3 and n = 18

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	3.5388739E-04	BLUE	0.1050092
ADMD1	4.7279926E-04	KSMD2	0.1183296
ADMD2	5.8612920E-04	ADMD1	0.1190109
CVMD1	6.7394995E-04	CVMD1	0.1217095
KSMD1	7.9139235E-04	CVMD2	0.1237090
CVMD2	8.1001956E-04	KSMD1	0.1239811
KSMD2	1.0424773E-03	ADMD2	0.1250392

TABLE XVI

Mean square error for c = 4 and n = 6

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	2.2004284E-03	BLUE	0.2905423
ADMD1	2.3699524E-03	KSMD2	0.3169968
CVMD1	2.5671816E-03	CVMD2	0.3370034
CVMD2	2.6604387E-03	KSMD1	0.3809848
KSMD1	2.7944316E-03	CVMD1	0.3836123
ADMD2	3.0173361E-03	ADMD2	0.3902711
KSMD2	3.1544713E-03	ADMD1	0.3999490

TABLE XVII

Mean square error for c = 4 and n = 9

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	9.8934094E-04	BLUE	0.1908786
ADMD1	1.1494281E-03	KSMD2	0.2049502
CVMD1	1.3984025E-03	CVMD2	0.2244719
ADMD2	1.4030078E-03	ADMD2	0.2334024
CVMD2	1.4857060E-03	ADMD1	0.2394877
KSMD1	1.5590133E-03	CVMD1	0.2420758
KSMD2	1.7792400E-03	KSMD1	0.2460075

TABLE XVIII

Mean square error for c = 4 and n = 12

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	4.3847732E-04	BLUE	0.1454246
ADMD1	5.3575670E-04	KSMD2	0.1606177
ADMD2	6.4326968E-04	ADMD2	0.1745982
CVMD1	7.0931221E-04	CVMD2	0.1752579
CVMD2	8.0394966E-04	ADMD1	0.1773998
KSMD1	8.4389572E-04	CVMD1	0.1846511
KSMD2	9.7926683E-04	KSMD1	0.1893843

TABLE XIX

Mean square error for c = 4 and n = 15

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	2.8805208E-04	BLUE	0.1210762
ADMD1	3.7486354E-04	KSMD2	0.1318542
ADMD2	4.3274325E-04	CVMD2	0.1393966
CVMD1	5.3938437E-04	ADMD2	0.1417205
CVMD2	6.0927385E-04	ADMD1	0.1437690
KSMD1	6.2738883E-04	CVMD1	0.1477181
KSMD2	7.4160466E-04	KSMD1	0.1574258

TABLE XX

Mean square error for c = 4 and n = 18

LOCATION (a)		SCALE (b)	
ESTIMATION TECHNIQUE	MSE	ESTIMATION TECHNIQUE	MSE
BLUE	1.9949843E-04	BLUE	9.3522320E-02
ADMD1	2.6885197E-04	KSMD2	1.0103040E-01
ADMD2	2.9941079E-04	ADMD1	1.0706935E-01
CVMD1	4.2770977E-04	ADMD2	1.0771950E-01
CVMD2	4.5176961E-04	CVMD2	1.0861484E-01
KSMD1	5.0622342E-04	CVMD1	1.1239370E-01
KSMD2	5.9582230E-04	KSMD1	1.1357192E-01

Appendix B

Computer Program for Estimator Comparison

The following FORTRAN computer program, BLUMD.FOR, was written to perform the Monte Carlo analysis and to generate the mean square errors for each estimation technique investigated. Program documentation is included within the program as comment statements to inform the reader of the purpose of each statement or group of statements. Additionally, each subroutine is prefaced by extensive documentation to inform the reader of the purpose of the subroutine, all of the variables used in the subroutine, the input variables required, the output variables generated, and the major computations performed within the subroutine to obtain the desired outputs.

```
c       BLUMD (BLU/MINIMUM DISTANCE) MAIN PROGRAM
c-----------------------------------------------------------------------
c Purpose:  BLUMD calculates the best linear unbiased estimates in
c           addition to the six minimum distance estimates
c           (based on the Kolmogorov, the Cramer-von Mises, and the
c           Anderson-Darling distances) for both the location and
c           scale parameters of the three parameter Pareto
c           distribution, where the shape parameter is varied
c           between the integer values 1,2,3 and 4.  Sample sizes
c           of 6, 9, 12, 15, and 18 are used.  Pareto variates are
c           generated for each combination of shape parameter and
c           sample size.  Finally, BLUMD calculates the mean square
c           error for each estimate type to compare which
c           estimation technique performs best.
c-----------------------------------------------------------------------
c Variables:      n = sample size
c                 c = shape parameter
c                nn = sample size symbol (varies from 1-5
c                     representing each permissable sample size)
c               kkk = dummy variable used to convert nn to n by
c                     using the formula:  kkk = 3 + (nn*3)
c             dseed = double precision seed for the Pareto variates
c                 x = array of Pareto variates
c                 B = array of B values used to calculate the blues
c                     for shape greater than 2
c                BB = array of BB values used to calculate the blues
c                     for shape less than or equal to 2
c                 D = constant used to calculate the blues for shape
c                     greater than 2
c               Anc = constant used to calculate the blues for shape
c                     less than or equal to 2
c               Bnc = constant used to calculate the blues for shape
c                     less than or equal to 2
c              ablu = blu estimate of location, a
c              bblu = blu estimate of scale, b
c               aKS = Kolmogorov minimum distance estimate for a
c               bKS = Kolmogorov minimum distance estimate for b
c              aCVM = Cramer-von Mises min distance estimate for a
c              bCVM = Cramer-von Mises min distance estimate for b
c               aAD = Anderson-Darling min distance estimate for a
c                     while holding  b = bblu  as constant
c               bAD = Anderson-Darling min distance estimate for b
c                     while holding  a = ablu  as constant
c              a2AD = Anderson-Darling min distance estimate for a
c              b2AD = Anderson-Darling min distance estimate for b
c              a1CV = Cramer-von Mises min distance estimate for a
c                     while holding  b = bblu  as constant
c              b1CV = Cramer-von Mises min distance estimate for b
c                     while holding  a = ablu  as constant
c              a1KS = Kolmogorov minimum distance estimate for a
c                     while holding  b = bblu  as constant
c              b1KS = Kolmogorov minimum distance estimate for b
c                     while holding  a = ablu  as constant
```

```
c              sse = array of sum of squared errors for each
c                    estimation technique where the true value of
c                    a is 1 and the true value of b is 1.
c              mse = array of mean square errors for each
c                    estimation technique used
c            count = array of counters used to count the number of
c                    valid estimate values found
c             anda = array of calculated A-D distance measures
c                    when estimating location alone
c             esta = array of location estimates used to minimize
c                    the A-D distance
c             icnt = counter for the number of location estimates
c                    used to minimize the A-D distance
c             andb = array of calculated A-D distance measures
c                    when estimating scale alone
c             estb = array of scale estimates used to minimize
c                    the A-D distance
c            icntb = counter for the number of scale estimates
c                    used to minimize the A-D distance
c            andab = array of calculated A-D distance measures
c                    when estimating  a  and  b  simultaneously
c             estaa = array of location estimates used to minimize
c                    the A-D distance
c             estbb = array of scale estimates used to minimize
c                    the A-D distance
c           icntab = counter for the number of location and scale
c                    estimates used to minimize the A-D distance
c-----------------------------------------------------------------------
c Inputs:    dseed = double precision seed for Pareto variate
c                    generation
c                c = shape parameter
c                n = sample size
c-----------------------------------------------------------------------
c Outputs:     mse = array of mean square errors for each
c                    estimation technique for each parameter under
c                    investigation (location and scale)
c-----------------------------------------------------------------------
c Calculate:   mse = sse/number of trials
c-----------------------------------------------------------------------
c       *** Variable Declarations
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1      aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1      ,andb,estb,icntb,andab,estaa,estbb,icntab
     1      ,aiCV,biCV,aiKS,biKS
        integer n,nn,count(4,5,14),c,kkk,icnt,icntb,icntab
        real x(18),ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,
     1      aCVM,bCVM,aAD,bAD,sse(4,5,14),mse(4,5,14),aZAD,bZAD
     1      ,anda(500),andb(500),andab(500),esta(500),estb(500),
     1       estaa(500),estbb(500),aiCV,biCV,aiKS,biKS
        double precision dseed
        call uerset(0,levold)
        dseed = 43846219217.d00
        print*,'dseed = ',dseed
```

```
      do 90 i=1,4
         c = i
         nn = 0
         do 80 j=6,18,3
            n = j
            nn = nn + 1
            do 40 jjj=1,14
               sse(c,nn,jjj) = 0
40          continue
            do 70 it = 1,1000
              if ((it.eq.200) .or. (it.eq.400) .or. (it.eq.600) .or.
     1           (it.eq.800) .or. (it.eq.1000))  then
                 print*,'c=',c,'  n=',n,'  iteration=',it
              end if
              call PARVAR
                 if (c .gt. 2) then
                    call BCGT2
                    call BLCGT2
                    go to 45
                 end if
              call BCLE2
              call BLCLE2
45               if (ablu .eq. 0 .and. bblu .eq. 0) then
                    go to 70
                 end if
              call KSMD
              call CVMMD
              call ADMD
              call ADBMD
              call AD2MD
              call CVAMD
              call CVBMD
              call KSAMD
              call KSBMD
c       *** Calculate the Sum of Squared Errors
58                sse(c,nn,1) = sse(c,nn,1) + (ablu - 1)**2
                  sse(c,nn,2) = sse(c,nn,2) + (bblu - 1)**2
                  sse(c,nn,3) = sse(c,nn,3) + (aKS - 1)**2
                  sse(c,nn,4) = sse(c,nn,4) + (bKS - 1)**2
                  sse(c,nn,5) = sse(c,nn,5) + (aCVM - 1)**2
                  sse(c,nn,6) = sse(c,nn,6) + (bCVM - 1)**2
                  sse(c,nn,7) = sse(c,nn,7) + (aAD - 1)**2
                  sse(c,nn,8) = sse(c,nn,8) + (bAD - 1)**2
                  sse(c,nn,9) = sse(c,nn,9) + (a2AD - 1)**2
                  sse(c,nn,10) = sse(c,nn,10) + (b2AD - 1)**2
                  sse(c,nn,11) = sse(c,nn,11) + (a1CV - 1)**2
                  sse(c,nn,12) = sse(c,nn,12) + (b1CV - 1)**2
                  sse(c,nn,13) = sse(c,nn,13) + (a1KS - 1)**2
                  sse(c,nn,14) = sse(c,nn,14) + (b1KS - 1)**2
                   if (it .eq. 1000) then
c       *** Calculate the mean square error for each estimate type
                     do 60 ll = 1,14
                       kkk = 3 + (nn*3)
```

```
                        if (count(c,nn,ll) .eq. 0) then
                         print*,'count=0 for c=',c,'  n=',kkk,'est=',ll
                         go to 60
                        end if
                        mse(c,nn,ll) = sse(c,nn,ll)/count(c,nn,ll)
                print*,'mse=',mse(c,nn,ll),'  c=',c,'  n=',kkk,' est=',ll
                print*,'count = ',count(c,nn,ll)
60                    continue
                    end if
70               continue
80             continue
90         continue
           end
           Subroutine PARVAR
c-------------------------------------------------------------------
c Purpose:  For a specified sample size, n, PARVAR
c           generates n random variates from a Pareto
c           distribution with location and scale parameters
c           set equal to one and the shape parameter, c,
c           set to either 1,2,3 or 4.
c-------------------------------------------------------------------
c Formula:        x = (1/r)**(1/c)
c-------------------------------------------------------------------
c Variables:
c                 r = random number
c                 c = shape parameter
c                 x = array of Pareto variates
c                 n = sample size
c             dseed = random number seed
c-------------------------------------------------------------------
c Inputs:     dseed = random number seed
c                 c = shape parameter
c                 n = sample size
c-------------------------------------------------------------------
c Outputs:        x = array of Pareto random variates
c-------------------------------------------------------------------
c Calculate:
c              x(j) = (1/r(j)) ** (1/c)
c
c-------------------------------------------------------------------
c       *** Variable Declarations
        real r(18),x(18),ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,a2AD,b2AD
        integer n,c,nn,count(4,5,14)
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1         aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD
        double precision dseed
        do 10 j=1,n
c       *** Call IMSL random number generator subroutine ggubs
           call ggubs(dseed,n,r)
c       *** Use the inverse transform technique for Pareto variates
           x(j)=(1/r(j))**(1/real(c))
10      continue
```

```
c       *** Call VSRTA to sort the variates in ascending order
        call vsrta(x,n)
        return
        end
        Subroutine BCLE2
c-----------------------------------------------------------------------
c Purpose:  For a given sample size, n, and a specified shape
c           (c=1 or c=2); BCLE2 calculates the B values used
c           to find the blu estimates of location and scale.
c           In addition, it calculates the constants Anc and Bnc
c           for the given shape and sample size.
c-----------------------------------------------------------------------
c Variables:  c = shape parameter
c             n = sample size
c            BB = array of B values  (k in number)
c             k = number of order statistics used; k=n-[2/c]
c            nc = product of n and c
c           Anc = constant in the formula for the blu for scale
c           Bnc = constant in the formula for the blu for scale
c-----------------------------------------------------------------------
c Inputs:     c = shape parameter
c             n = sample size
c-----------------------------------------------------------------------
c Outputs:   BB = array of B values
c           Anc = constant in the blue for the scale parameter
c           Bnc = constant in the blue for the scale parameter
c-----------------------------------------------------------------------
c Calculate:
c           Anc = (c+1)(c+2)(nc-1) / (nc-2)(nc-c-2)
c
c           Bnc = (nc-2) / (c+2)
c
c          For c=1 :    B(i) = (1 - i/n) [ 1 - i/(n-1) ]
c
c          For c=2 :    B(i) = 1 - i/n
c
c-----------------------------------------------------------------------
c       *** Variable Declarations:
        real nc,x(18),ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,aZAD,bZAD
        integer n,k,c,nn,count(4,5,14)
        double precision dseed
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1         aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD
        k=n-(2/c)
c       *** Calculate the B values when c=1
          if (c.eq.1) then
             do 10 j=1,k
                BB(j)=(1-j/real(n))*(1-j/(real(n)-1))
10           continue
             go to 36
          end if
c       *** Calculate the B values when c=2 (i.e., c.ne.1)
```

```
      do 20 m=1,k
         BB(m)=1-m/real(n)
20    continue
36    nc = n * c
c     *** Calculate the constants Anc and Bnc
      Anc= ((c+1)*(c+2)*(nc-1)) / ((nc-2)*(nc-c-2))
      Bnc= (nc-2) / (c+2)
      return
      end
      Subroutine BLCLE2
c-----------------------------------------------------------------------
c Purpose:  For a given sample size, n, and shape (c=1 or c=2),
c           BLCLE2 calculates the best linear unbiased estimates
c           of location and scale for a sample of ordered Pareto
c           variates.
c-----------------------------------------------------------------------
c Variables:  x = array of ordered Pareto variates
c             c = shape parameter
c             n = sample size
c            BB = array of B values used to calculate the blues
c            nc = product of n and c
c             k = number of order statistics used; k = n - [2/c]
c           Anc = constant used to calculate the blue for scale
c           Bnc = constant used to calculate the blue for scale
c         Bxsum = sum of [B(i) * x(i)] terms for i = 1,2,. . . ,k
c          ablu = blu estimate of the location parameter, a
c          bblu = blu estimate of the scale parameter, b
c         count = array of counters used to count the number of
c                 valid estimate values found
c-----------------------------------------------------------------------
c Inputs:     x = array of ordered Pareto variates
c             c = shape parameter
c             n = sample size
c            BB = array of B values used to calculate the blues
c           Anc = constant used to calculate the blue for scale
c           Bnc = constant used to calculate the blue for scale
c-----------------------------------------------------------------------
c Outputs: ablu = blu estimate of the location parameter, a
c          bblu = blu estimate of the scale parameter, b
c-----------------------------------------------------------------------
c Calculate:
c             b = (Anc) [ B(1)x(1) - B(2)x(2) + ... + B(k)x(k)
c                       - (Bnc)x(1) ]
c
c             a = x(1) - [ b/(nc-1) ]
c
c-----------------------------------------------------------------------
c     *** Variable Declarations:
      real x(18),ablu,bblu,Bxsum,nc,B(18),D,Anc,Bnc,BB(18),
     1     aKS,bKS,aCVM,bCVM,aAD,bAD,aZAD,bZAD
      integer n,k,c,nn,count(4,5,14)
      double precision dseed
      common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
```

```
     1        aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD
          Bxsum=0
          k=n-(2/c)
c         *** Sum the products of B(i) and x(i) for i = 1,2,...,k
          do 10 j=1,k
             Bxsum=Bxsum+BB(j)*x(j)
10        continue
          nc=n*c
c         *** Calculate the blue for scale, then for location
          bblu=Anc*(Bxsum - Bnc*x(1))
          ablu=x(1)-bblu/(nc-1)
c         *** Increment counter for valid blues
             if (bblu .gt. 0) then
                count(c,nn,1) = count(c,nn,1) + 1
                count(c,nn,2) = count(c,nn,2) + 1
             else
                print*,'bblu=',bblu,'ablu=',ablu,'  negativity'
                ablu = 0
                bblu = 0
             end if
          return
          end
          Subroutine BCGT2
c-------------------------------------------------------------------
c Purpose:  For a given sample size, n, and a specified shape,
c           c)2; BCGT2 calculates the B and D values used to
c           find the blu estimates of location and scale.
c-------------------------------------------------------------------
c Variables:    c = shape parameter
c               n = sample size
c               B = array of B values (n in size)
c               D = D value
c            bsum = sum of B values for i=1 . . . (n-1)
c-------------------------------------------------------------------
c Inputs:       c = shape parameter
c               n = sample size
c-------------------------------------------------------------------
c Outputs:      B = array of B values
c               D = D value
c-------------------------------------------------------------------
c Calculate:
c            B(i) = [1 - 2/c(n-i+1)] * B(i-1)
c
c                D = (c+1)[B(1) + B(2) + ... + B(n-1)] + (c-1)B(n)
c
c-------------------------------------------------------------------
c         *** Variable Declarations:
          real bsum,x(18),ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,
     1         aCVM,bCVM,aAD,bAD,a2AD,b2AD
          integer n,c,nn,count(4,5,14)
          double precision dseed
          common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1         aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD
```

```
c       *** Calculate the first B value
        B(1)=(1-(2/(c*real(n))))
c       *** Calculate the second thru the nth B values
        do 10 j=2,n
           B(j)=B(j-1)*(1-(2/(real(c)*(n-j+1))))
10      continue
        bsum=0
c       *** Sum the 'first' to the 'nth minus one' B values
        do 20 k=1,(n-1)
           bsum=bsum+B(k)
20      continue
c       *** Calculate the D value
        D = (c+1) * bsum  +  (c-1) * B(n)
        return
        end
        Subroutine BLCGT2
c-----------------------------------------------------------------------
c Purpose:  For a given sample size, n, and a specified shape,
c           c>2, BLCGT2 calculates the best linear unbiased
c           estimates of location and scale.
c-----------------------------------------------------------------------
c Variables:  x = array of ordered Pareto variates
c             c = shape parameter
c             n = sample size
c             B = array of B values used to calculate the blues
c             D = D value used to calculate the blues
c             Y = Y value used to calculate the blues
c          ablu = blu for location parameter, a
c          bblu = blu for scale parameter, b
c         Bxsum = sum of [B(i) * x(i)] terms for i = 1,2...,(n-1)
c            nc = product of n and c
c         count = array of counters used to count the number of
c                 valid estimate values found
c-----------------------------------------------------------------------
c Inputs:     x = array of ordered Pareto variates
c             c = shape parameter
c             n = sample size
c             B = array of B values used to calculate blues
c             D = D value used to calculate the blues
c-----------------------------------------------------------------------
c Outputs: ablu = blu estimate for location, a
c          bblu = blu estimate for scale, b
c-----------------------------------------------------------------------
c Calculate:
c             Y = (c+1)[ B(1)x(1) + B(2)x(2) + ... + B(n-1)x(n-1) ]
c                 + (c-1)[ B(n)x(n) ] - Dx(1)
c
c             a = x(1) - Y/[(nc-1)(nc-2) - Dnc]
c
c             b = (nc-1) [ x(1) - a ]
c
c-----------------------------------------------------------------------
c       *** Variable Declarations:
```

```
      real x(18),ablu,bblu,Bxsum,nc,Y,B(18),D,Anc,Bnc,BB(18)
     1    ,aKS,bKS,aCVM,bCVM,aAD,bAD,aZAD,bZAD
      integer n,c,nn,count(4,5,14)
      double precision dseed
      common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD
      Bxsum=0
c     *** Sum the products of the B(i) and x(i) values to i = n-1
      do 10 j=1,(n-1)
         Bxsum=Bxsum+B(j)*x(j)
10    continue
c     *** Calculate the Y value
      Y=(c+1)*Bxsum+(c-1)*B(n)*x(n)-D*x(1)
      nc=n*c
c     *** Calculate the blu estimates for location and scale
      ablu=x(1)-Y/((nc-1)*(nc-2)-(nc*D))
      bblu=(nc-1)*(x(1)-ablu)
c     *** Increment counters for valid blues
         if (bblu .gt. 0) then
            count(c,nn,1) = count(c,nn,1) + 1
            count(c,nn,2) = count(c,nn,2) + 1
         else
            print*,'ablu=',ablu,'   bblu=',bblu,'   negativity'
            ablu = 0
            bblu = 0
         end if
      return
      end
      Subroutine KSMD
c-----------------------------------------------------------------------
c Purpose:  KSMD generates the minimum distance estimates of
c           location and scale based upon minimizing the
c           Kolmogorov distance measure defined in subroutine
c           KDIS.  This routine uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Kolmogorov distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of Kolmogorov distance at the final
c                   parameter estimates
c             kse = Kolmogorov derived minimum distance estimates
c             aKS = Kolmogorov minimum distance location estimate
c             bKS = Kolmogorov minimum distance scale estimate
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c           count = array of counters used to count the number of
c                   valid estimate values found
c-----------------------------------------------------------------------
```

```
c Inputs:    NPAR = number of parameters altered while minimizing
c            NSIG = number of significant digits required
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c             kse = initial estimates for the minimization process
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c-----------------------------------------------------------------
c Outputs:     F = minimum value of the function being minimized
c             kse = revised estimate values
c             aKS = revised MD estimate of location  [aKS = kse(1)]
c             bKS = revised MD estimate of scale     [bKS = kse(2)]
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c-----------------------------------------------------------------
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1         aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD
        external kdis
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14)
        real kse(2),H(3),G(2),W(6),F,x(18),ablu,bblu,aKS,bKS,
     1      B(18),D,Anc,Bnc,BB(18),aCVM,bCVM,aAD,bAD,a2AD,b2AD
        double precision dseed
c       *** Enter the ZXMIN required constants
        NPAR = 2
        NSIG = 3
        MAXFN = 500
        IOPT = 0
c       *** Initialize the kse values to the blu estimates
        kse(1) = ablu
        kse(2) = bblu
c       *** Call ZXMIN to refine the kse values by minimizing
c       ***      the Kolmogorov distance (KST) computed in the
c       ***      subroutine KDIS
        call ZXMIN(KDIS,NPAR,NSIG,MAXFN,IOPT,kse,H,G,F,W,IER)
c       *** Relabel the refined estimates of location and scale
        aKS = kse(1)
        bKS = kse(2)
c       *** Increment the KS counters
        count(c,nn,3) = count(c,nn,3) + 1
        count(c,nn,4) = count(c,nn,4) + 1
        return
        end
        Subroutine KDIS(NPAR,kse,F)
c-----------------------------------------------------------------
c Purpose:  KDIS provides the function which is to be minimized
c           by ZXMIN for the Kolmogorov distance measure.  The
c           location and scale parameters are altered to achieve
c           this minimization.
c-----------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
```

```
c                n = sample size
c              kse = estimates of the parameters being altered
c                F = value of the function to be minimized
c                x = array of ordered Pareto variates
c                c = shape parameter
c               z1 = array of Pareto cdf points
c               DP = positive differences between the EDF and cdf
c               DM = negative differences between the EDF and cdf
c            DPLUS = maximum positive difference
c           DMINUS = maximum negative difference
c              KST = Maximum of DPLUS and DMINUS
c-----------------------------------------------------------------
c Inputs:     NPAR = number of parameters available to alter
c                n = sample size
c              kse = initial estimates (the blu estimates)
c                x = array of ordered Pareto variates
c                c = shape parameter
c-----------------------------------------------------------------
c Outputs:       F = value of the function at the final estimates
c              kse = revised estimates of location and scale;
c                    these are the Kolmogorov minimum distance
c                    estimates
c-----------------------------------------------------------------
c Calculations:
c             z(1) = 1 - (1 + [x(1)-a]/b)**(-c)
c
c            DP(i) = ABS[ 1/n - z(i) ]
c
c            DM(i) = ABS[ z(1) - (1-1)/n ]
c
c-----------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1          aCVM,bCVM,aAD,Bad,nn,count,a2AD,b2AD
        integer NPAR,n,c,nn,count(4,5,14)
        real kse(NPAR),F,x(18),z1(18),DP(18),DM(18),DPLUS,
     1          DMINUS,KST,ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,
     1          aCVM,bCVM,aAD,bAD,a2AD,b2AD
        double precision dseed
c       *** Calculate the Pareto cdf value [z1(j)] at each point
c       ***      and the differences between the EDF step function
c       ***      and the cdf points
        do 10 j=1,n
           zi(j) = 1-(1/(1+(x(j)-kse(1))/kse(2)))**c
           DP(j) = ABS(j/real(n) - z1(j))
           DM(j) = ABS(z1(j) - (j-1)/real(n))
10      continue
c       *** Select the maximum of the plus and minus differences
        DPLUS = MAX(DP(1),DP(2),DP(3),DP(4),DP(5),DP(6),DP(7)
     1      ,DP(8),DP(9),DP(10),DP(11),DP(12),DP(13),DP(14)
     1      ,DP(15),DP(16),DP(17),DP(18))
        DMINUS = MAX(DM(1),DM(2),DM(3),DM(4),DM(5),DM(6),DM(7)
     1      ,DM(8),DM(9),DM(10),DM(11),DM(12),DM(13),DM(14)
```

```
     1      ,DM(15),DM(16),DM(17),DM(18))
c       *** Select the maximum Kolmogorov distance measure and
c       ***     set F equal to that distance.  F becomes the
c       ***     function which ZXMIN attempts to minimize by
c       ***     altering the values of the location and scale
c       ***     parameters
        KST = MAX(DPLUS,DMINUS)
        F = KST
        return
        end
        Subroutine CVMMD
c-----------------------------------------------------------------------
c Purpose:  CVMMD generates the minimum distance estimates of
c           location and scale based upon minimizing the
c           Cramer-von Mises distance measure defined in subroutine
c           CVMDIS.  This routine uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Cramer-von Mises (CVM) distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of CVM distance at the final
c                   parameter estimates
c            cvme = CVM derived minimum distance estimates
c            aCVM = CVM minimum distance location estimate
c            bCVM = CVM minimum distance scale estimate
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c           count = array of counters used to count the number of
c                   valid estimate values found
c-----------------------------------------------------------------------
c Inputs:    NPAR = number of parameters altered while minimizing
c            NSIG = number of significant digits required
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c            cvme = initial estimates for the minimization process
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c-----------------------------------------------------------------------
c Outputs:      F = minimum value of the function being minimized
c            cvme = revised estimate values
c            aCVM = revised MD estimate of location  [aCVM=cvme(1)]
c            bCVM = revised MD estimate of scale     [bCVM=cvme(2)]
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c
c-----------------------------------------------------------------------
```

```
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1          aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD
        external cvmdis
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14)
        real cvme(2),H(3),G(2),W(6),F,x(18),ablu,bblu,aCVM,bCVM
     1          ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aAD,bAD,a2AD,b2AD
        double precision dseed
c       *** Enter the ZXMIN required constants
        NPAR = 2
        NSIG = 3
        MAXFN = 500
        IOPT = 0
c       *** Initialize the cvme values to the blu estimates
        cvme(1) = ablu
        cvme(2) = bblu
c       *** Call ZXMIN to refine the cvme values by minimizing
c       ***      the CVM distance (W2) computed in the
c       ***      subroutine CVMDIS
        call ZXMIN(CVMDIS,NPAR,NSIG,MAXFN,IOPT,cvme,H,G,F,W,IER)
c       *** Relabel the refined estimates of location and scale
        aCVM = cvme(1)
        bCVM = cvme(2)
c       *** Increment the CVM counters
        count(c,nn,5) = count(c,nn,5) + 1
        count(c,nn,6) = count(c,nn,6) + 1
        return
        end
        Subroutine CVMDIS(NPAR,cvme,F)
c-----------------------------------------------------------------------
c Purpose:  CVMDIS provides the function which is to be minimized
c           by ZXMIN for the Cramer-von Mises distance measure.
c           The location and scale parameters are altered to
c           achieve this minimization.
c-----------------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
c                 n = sample size
c              cvme = estimates of the parameters being altered
c                 F = value of the function to be minimized
c                 x = array of ordered Pareto variates
c                 c = shape parameter
c                zi = array of Pareto cdf points
c               ACV = the squared quantity in the W2 formula
c              SCVM = the sum of the ACV quantities
c                W2 = the CVM distance measure
c-----------------------------------------------------------------------
c Inputs:      NPAR = number of parameters available to alter
c                 n = sample size
c              cvme = initial estimates (the blu estimates)
c                 x = array of ordered Pareto variates
c                 c = shape parameter
```

```
c-----------------------------------------------------------------------
c Outputs:        F = value of the function at the final estimates
c              cvme = revised estimates of location and scale;
c                     these are the CVM minimum distance estimates
c-----------------------------------------------------------------------
c Calculations:
c              z(i) = 1 - (1 + [x(i)-a]/b)**(-c)
c
c            ACV(i) = [ z(i) - (2i-1)/2n ]**2
c
c              SCVM = ACV(1) + ACV(2) + ... + ACV(n)
c
c                W2 = SCVM + 1/12n
c
c-----------------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1     aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD
        integer NPAR,n,c,nn,count(4,5,14)
        real cvme(NPAR),F,x(18),zi(18),SCVM,ACV(18),W2
     1     ,ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,
     1     aAD,bAD,aZAD,bZAD
        double precision dseed
        SCVM = 0
        do 10 j=1,n
           zi(j) = 1-(1/(1+(x(j)-cvme(1))/cvme(2)))**c
           ACV(j) = (zi(j) - (2*j-1)/(2*real(n)))**2
           SCVM = SCVM + ACV(j)
10      continue
        W2 = SCVM + (1/(12*real(n)))
        F = W2
        return
        end
        Subroutine ADMD
c-----------------------------------------------------------------------
c Purpose:  ADMD generates the minimum distance estimates of
c           the location parameter based upon minimizing the
c           Anderson-Darling distance measure defined in
c           subroutine ADDIS.  ADMD uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Anderson-Darling (A-D) distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of A-D distance at the final
c                   parameter estimates
c             ade = A-D derived minimum distance estimate
```

```
c             aAD = A-D minimum distance location estimate
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c           count = array of counters used to count the number of
c                   valid estimate values found
c            anda = array of calculated A-D distance measures
c            esta = array of location estimates used to minimize
c                   the A-D distance measure
c            icnt = counter for the number of estimates used
c-----------------------------------------------------------------
c Inputs:    NPAR = number of parameters altered while minimizing
c            NSIG = number of significant digits required
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c             ade = initial estimate for the minimization process
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c-----------------------------------------------------------------
c Outputs:     F = minimum value of the function being minimized
c             ade = revised estimate values
c             aAD = revised MD estimate of location  [aAD = ade(1)]
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c-----------------------------------------------------------------
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1       ,andb,estb,icntb,andab,estaa,estbb,icntab
        external addis
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1       ,icntb,icntab
        real ade(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1       ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,aZAD,bZAD
     1       ,anda(500),esta(500),AD1,andb(500),estb(500),ardab(500)
     1       ,estaa(500),estbb(500)
        double precision dseed
c       *** Enter the ZXMIN required constants
        NPAR = 1
        NSIG = 3
        MAXFN = 500
        IOPT = 0
c       *** Initialize the ade value to the blu estimate
        ade(1) = ablu
c       *** Call ZXMIN to refine the ade values by minimizing
c       ***      the Anderson-Darling distance (AD) computed in
c       ***      the subroutine ADDIS
        call ZXMIN(ADDIS,NPAR,NSIG,MAXFN,IOPT,ade,H,G,F,W,IER)
        aAD = ade(1)
c       *** Reinitialize the icnt, anda, and esta arrays
```

```
25      icnt = 0
        do 30 i = 1,W(2)
           anda(i) = 0
           esta(i) = 0
30      continue
c       *** Increment AD counter for valid AD estimates
        count(c,nn,7) = count(c,nn,7) + 1
        return
        end
        Subroutine ADDIS(NPAR,ade,F)
c-----------------------------------------------------------------
c Purpose:  ADDIS provides the function which is to be minimized
c           by ZXMIN for the Anderson-Darling distance measure.
c           The location parameter is altered to achieve the
c           minimization.
c-----------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
c                n = sample size
c              ade = estimates of the parameter being altered
c                F = value of the function to be minimized
c                x = array of ordered Pareto variates
c                c = shape parameter
c               zi = array of Pareto cdf points
c              AAA = array of terms to be summed in AD formula
c             SAAD = sum of the AAA(i) terms
c               AD = Anderson-Darling distance measure
c             anda = array of calculated A-D distance measures
c             esta = array of location estimates used to minimize
c                    the A-D distance measure
c             icnt = counter for the number of estimates used
c-----------------------------------------------------------------
c Inputs:      NPAR = number of parameters available to alter
c                n = sample size
c              ade = initial estimate (the blu estimate)
c                x = array of ordered Pareto variates
c                c = shape parameter
c-----------------------------------------------------------------
c Outputs:       F = value of the function at the final estimate
c              ade = revised estimate of location; this is the
c                    Anderson-Darling minimum distance estimate
c-----------------------------------------------------------------
c Calculations:
c              z(i) = 1 - (1 + [x(i)-a]/b)**(-c)
c
c            AAA(i) = (2i-1) [ ln z(i)  +  ln ( 1-z(n+1-i) ) ]
c
c              SAAD = AAA(1) + AAA(2) + ... + AAA(n)
c
c                AD = (-SAAD)/n - n
c
c-----------------------------------------------------------------
```

```
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1     aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD,anda,esta,icnt
     1     ,andb,estb,icntb,andab,estaa,estbb,icntab
        integer NPAR,n,c,nn,count(4,5,14),icnt,icntb,icntab
        real ade(NPAR),F,x(18),zi(18),AAA(18),SAAD,AD
     1     ,ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,
     1      aAD,bAD,a2AD,b2AD,anda(500),esta(500)
     1     ,andb(500),estb(500),ancab(500),estaa(500),estbb(500)
        double precision dseed
        do 10 j=1,n
c       *** Calculate the Pareto cdf point values
           zi(j) = 1-(1/(1+(x(j)-ade(1))/bblu))**c
c       *** Test zi(j) and [ 1 - zi(j) ] for negativity
           if (zi(j).le.0 .or. zi(j).ge.1) then
              go to 30
           end if
10      continue
        SAAD = 0
c       *** Calculate the Anderson-Darling distance
        do 20 m=1,n
           AAA(m) = (2*m-1) * (log(zi(m)) + log(1-zi(n+1-m)))
           SAAD = SAAD + AAA(m)
20      continue
        AD = (-1) * (n + SAAD/n)
c       *** Save the AD and ade(1) values
        icnt = icnt + 1
        anda(icnt) = AD
        esta(icnt) = ade(1)
c       *** Relabel the A-D distance
        F = AD
        go to 40
30      ade(1) = esta(icnt-1)
40      return
        end
        Subroutine ADBMD
c-----------------------------------------------------------------------
c Purpose:  ADBMD generates the minimum distance estimates of
c           the scale parameter based upon minimizing the
c           Anderson-Darling distance measure defined in
c           subroutine ADBDIS.  ADBMD uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Anderson-Darling (A-D) distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of A-D distance at the final
```

```
c                    parameter estimates
c              ade = A-D derived minimum distance estimate
c              bAD = A-D minimum dis'ance scale estimate
c             ablu = blu estimate of location
c             bblu = blu estimate of scale
c            count = array of counters used to count the number of
c                    valid estimate values found
c             andb = array of calculated A-D distance measures
c             estb = array of scale estimates used to minimize
c                    the A-D distance measure
c            icntb = counter for the number of estimates used
c-----------------------------------------------------------------
c Inputs:     NPAR = number of parameters altered while minimizing
c             NSIG = number of significant digits required
c            MAXFN = maximum number of function evaluations
c             IOPT = options selector (see IMSL manual on ZXMIN)
c              ade = initial estimate for the minimization process
c             ablu = blu estimate of location
c             bblu = blu estimate of scale
c-----------------------------------------------------------------
c Outputs:       F = minimum value of the function being minimized
c              ade = revised estimate values
c              bAD = revised MD estimate of scale  [bAD = ade(1)]
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c              IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c-----------------------------------------------------------------
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1      aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD,anda,esta,icnt
     1      ,andb,estb,icntb,andab,estaa,estbb,icntab
        external adbdis
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt
     1      ,icntb,icntab,hhh
        real ade(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1      ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,a2AD,b2AD
     i      ,anda(500),esta(500),andb(500),estb(500),andab(500),
     1       estaa(500),estbb(500),AD2
        double precision dseed
c       *** Enter the ZXMIN required constants
        NPAR = 1
        NSIG = 3
        MAXFN = 500
        IOPT = 0
c       *** Initialize the ade value to the blu estimate
        ade(1) = bblu
c       *** Call ZXMIN to refine the ade values by minimizing
c       ***      the Anderson-Darling distance (AD) computed in
c       ***      the subroutine ADBDIS
        call ZXMIN(ADBDIS,NPAR,NSIG,MAXFN,IOPT,ade,H,G,F,W,IER)
```

```
        bAD = ade(1)
c       *** Reinitialize the icntb, andb, and estb arrays
25      icntb = 0
        do 30 i = 1,W(2)
           andb(i) = 0
           estb(i) = 0
30      continue
c       *** Increment AD counter for valid AD estimates
        count(c,nn,8) = count(c,nn,8) + 1
        return
        end
        Subroutine ADBDIS(NPAR,ade,F)
c-----------------------------------------------------------------
c Purpose:  ADBDIS provides the function which is to be minimized
c           by ZXMIN for the Anderson-Darling distance measure.
c           The scale parameter is altered to achieve the
c           minimization.
c-----------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
c                n = sample size
c              ade = estimates of the parameter being altered
c                F = value of the function to be minimized
c                x = array of ordered Pareto variates
c                c = shape parameter
c               zi = array of Pareto ( 'f points
c              AAA = array of terms to be summed in AD formula
c             SAAD = sum of the AAA(i) terms
c               AD = Anderson-Darling distance measure
c             andb = array of calculated A-D distance measures
c             estb = array of scale estimates used to minimize
c                    the A-D distance measure
c            icntb = counter for the number of estimates used
c-----------------------------------------------------------------
c Inputs:      NPAR = number of parameters available to alter
c                n = sample size
c              ade = initial estimate (the blu estimate)
c                x = array of ordered Pareto variates
c                c = shape parameter
c-----------------------------------------------------------------
c Outputs:       F = value of the function at the final estimate
c              ade = revised estimate of scale; this is the
c                    Anderson-Darling minimum distance estimate
c-----------------------------------------------------------------
c Calculations:
c              z(i) = 1 - (1 + [x(i)-a]/b)**(-c)
c
c            AAA(i) = (2i-1) [ ln z(i)  +  ln ( 1-z(n+1-i) ) ]
c
c              SAAD = AAA(1) + AAA(2) + ... + AAA(n)
c
c                AD = (-SAAD)/n - n
```

```
c
c-----------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1     aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1     ,andb,estb,icntb,andab,estaa,estbb,icntab
        integer NPAR,n,c,nn,count(4,5,14),icnt,icntb,icntab
        real ade(NPAR),F,x(18),zi(18),AAA(18),SAAD,AD
     1     ,ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,
     1      aAD,bAD,aZAD,bZAD
     1     ,anda(500),esta(500),andb(500),estb(500),andab(500),
     1      estaa(500),estbb(500)
        double precision dseed
        do 10 j=1,n
c       *** Calculate the Pareto cdf point values
           zi(j) = 1-(1/(1+(x(j)-ablu)/ade(1)))**c
c       *** Test zi(j) and [ 1 - zi(j) ] for negativity
           if (zi(j).le.0 .or. zi(j).ge.1) then
              go to 30
           end if
10      continue
        SAAD = 0
c       *** Calculate the Anderson-Darling distance
        do 20 m=1,n
           AAA(m) = (2*m-1) * (log(zi(m)) + log(1-zi(n+1-m)))
           SAAD = SAAD + AAA(m)
20      continue
        AD = (-i) * (n + SAAD/n)
c       *** Save the AD and ade(i) values
        icntb = icntb + 1
        andb(icntb) = AD
        estb(icntb) = ade(i)
c       *** Relabel the A-D distance
        F = AD
        go to 40
30      ade(i) = estb(icntb-i)
40      return
        end
        Subroutine AD2MD
c-----------------------------------------------------------------
c Purpose-  AD2MD generates the minimum distance estimates of
c           location and scale simultaneously, based on minimizing
c           the Anderson-Darling distance measure defined in
c           subroutine AD2DIS.  AD2MD uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------
c Variables- NPAR - number of parameters altered by minimizing
c                   the Anderson-Darling (A-D) distance
c            NSIG - number of significant digits for convergence
c           MAXFN - maximum number of function evaluations
c            IOPT - options selector (see IMSL manual on ZXMIN)
```

```
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER - error parameter (see IMSL manual on ZXMIN)
c               F - value of A-D distance at the final
c                   parameter estimates
c             ade - A-D derived minimum distance estimate
c           aZAD - A-D minimum distance location estimate
c           bZAD - A-D minimum distance scale estimate
c           ablu - blu estimate of location
c           bblu - blu estimate of scale
c          count - array of counters used to count the number of
c                  valid estimate values found
c          andab - array of calculated A-D distance measures
c          estaa - array of location estimates used to minimize
c                  the A-D distance measure
c          estbb - array of scale estimates used to minimize
c                  the A-D distance measure
c         icntab - counter for the number of estimates used
c-----------------------------------------------------------------------
c Inputs.    NPAR - number of parameters altered while minimizing
c            NSIG - number of significant digits required
c           MAXFN - maximum number of function evaluations
c            IOPT - options selector (see IMSL manual on ZXMIN)
c             ade - initial estimate for the minimization process
c            ablu - blu estimate of location
c            bblu - blu estimate of scale
c-----------------------------------------------------------------------
c Outputs.     F - minimum value of the function being minimized
c             ade - revised estimate values
c           aZAD - revised MD estimate of location [aZAD - ade(1)]
c           bZAD - revised MD estimate of scale  [bZAD - ade(2)]
c         H, G, W - vectors defined in IMSL manual on ZXMIN
c             IER - error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------------
c Calculate.  no calculations performed in this subroutine
c-----------------------------------------------------------------------
c       *** Variables Declaration.
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     i       aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     i       ,andb,estb,icntb,andab,estaa,estbb,icntab
        external adZdis
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14)
     i       ,icnt,icntb,icntab,hhhh
        real ade(2),H(3),G(2),W(6),F,x(18),ablu,bblu,aAD,bAD
     i       ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,aZAD,bZAD
     i       ,anda(500),esta(500),andb(500),estb(500),andab(500),
     i        estaa(500),estbb(500),AD3
        double precision dseed
c       *** Enter the ZXMIN required constants
        NPAR = 2
        NSIG = 3
        MAXFN = 500
```

```
        IOPT = 0
c       *** Initialize the ade value to the blu estimate
        ade(1) = ablu
        ade(2) = bblu
c       *** Call ZXMIN to refine the ade values by minimizing
c       ***       the Anderson-Darling distance (AD) computed in
c       ***       the subroutine AD2DIS
        call ZXMIN(AD2DIS,NPAR,NSIG,MAXFN,IOPT,ade,H,G,F,W,IER)
        aZAD = ade(1)
        bZAD = ade(2)
c       *** Reinitialize the icntab, andab, estbb, and estaa arrays
25      icntab = 0
        do 30 i = 1,W(2)
           andab(i) = 0
           estaa(i) = 0
           estbb(i) = 0
30      continue
c       *** Increment AD counter for valid AD estimates
        count(c,nn,9) = count(c,nn,9) + 1
        count(c,nn,10) = count(c,nn,10) + 1
        return
        end
        Subroutine AD2DIS(NPAR,ade,F)
c-----------------------------------------------------------------
c Purpose:  AD2DIS provides the function which is to be minimized
c           by ZXMIN for the Anderson-Darling distance measure.
c           The location and scale parameters are both altered to
c           achieve the minimization.
c-----------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
c                n = sample size
c              ade = estimates of the parameter being altered
c                F = value of the function to be minimized
c                x = array of ordered Pareto variates
c                c = shape parameter
c               z1 = array of Pareto cdf points
c              AAA = array of terms to be summed in AD formula
c             SAAD = sum of the AAA(i) terms
c               AD = Anderson-Darling distance measure
c            andab = array of calculated A-D distance measures
c            estaa = array of location estimates used to minimize
c                    the A-D distance measure
c            estbb = array of scale estimates used to minimize
c                    the A-D distance measure
c           icntab = counter for the number of estimates used
c-----------------------------------------------------------------
c Inputs:      NPAR = number of parameters available to alter
c                n = sample size
c              ade = initial estimate (the blu estimate)
c                x = array of ordered Pareto variates
c                c = shape parameter
```

```
c-----------------------------------------------------------------------
c Outputs:        F = value of the function at the final estimate
c                ade = revised estimates of location and scale;
c                      these are the Anderson-Darling minimum
c                      distance estimates
c-----------------------------------------------------------------------
c Calculations:
c              z(i) = 1 - (1 + [x(i)-a]/b)**(-c)
c
c            AAA(i) = (2i-1) [ ln z(i)  +  ln ( 1-z(n+1-i) ) ]
c
c              SAAD = AAA(1) + AAA(2) + ... + AAA(n)
c
c                AD = (-SAAD)/n - n
c
c-----------------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1       ,andb,estb,icntb,andab,estaa,estbb,icntab
        integer NPAR,n,c,nn,count(4,5,14),icnt,icntb,icntab
        real ade(NPAR),F,x(18),z1(18),AAA(18),SAAD,AD
     1       ,ablu,bblu,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,
     1        aAD,bAD,a2AD,bZAD,anda(500),esta(500),andb(500),
     1        estb(500),andab(500),estaa(500),estbb(500)
        double precision dseed
        do 10 j=1,n
c       *** Calculate the Pareto cdf point values
           z1(j) = 1-(1/(1+(x(j)-ade(1))/ade(2)))**c
c       *** Test zi(j) and [ 1 - z1(j) ] for negativity
           if (z1(j).le.0 .or. z1(j).ge.1) then
              go to 30
           end if
10      continue
        SAAD = 0
c       *** Calculate the Anderson-Darling distance
        do 20 m=1,n
           AAA(m) = (2*m-1) * (log(z1(m)) + log(1-z1(n+1-m)))
           SAAD = SAAD + AAA(m)
20      continue
        AD = (-1) * (n + SAAD/n)
c       *** Save the AD and ade(1) values
        icntab = icntab + 1
        andab(icntab) = AD
        estaa(icntab) = ade(1)
        estbb(icntab) = ade(2)
c       *** Relabel the A-D distance
        F = AD
        go to 40
30      ade(1) = estaa(icntab-1)
        ade(2) = estbb(icntab-1)
40      return
        end
```

```
      Subroutine CVAMD
c-----------------------------------------------------------------------
c Purpose:  CVAMD generates the minimum distance estimates of
c           the location parameter based upon minimizing the
c           Cramer-von Mises distance measure defined in
c           subroutine CVADIS.  CVAMD uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Cramer-von Mises (CVM) distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of CVM distance at the final
c                   parameter estimates
c            cvme = CVM derived minimum distance estimate
c            aICV = CVM minimum distance location estimate
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c           count = array of counters used to count the number of
c                   valid estimate values found
c-----------------------------------------------------------------------
c Inputs:    NPAR = number of parameters altered while minimizing
c            NSIG = number of significant digits required
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c            cvme = initial estimate for the minimization process
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c-----------------------------------------------------------------------
c Outputs:      F = minimum value of the function being minimized
c            cvme = revised estimate values
c            aICV = revised MD estimate of location [aICV=cvme(1)]
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c-----------------------------------------------------------------------
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1      aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD,anda,esta,icnt
     1      ,andb,estb,icntb,andab,estaa,estbb,icntab
     1      ,aICV,bICV,aIKS,bIKS
        external CVADIS
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1      ,icntb,icntab
        real cvme(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1      ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,a2AD,b2AD
     1      ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
```

```
     1      ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
            double precision dseed
      c     *** Enter the ZXMIN required constants
            NPAR = 1
            NSIG = 3
            MAXFN = 500
            IOPT = 0
      c     *** Initialize the cvme value to the blu estimate
            cvme(1) = ablu
      c     *** Call ZXMIN to refine the cvme values by minimizing
      c     ***      the Cramer-von Mises distance (W2) computed in
      c     ***      the subroutine CVADIS
            call ZXMIN(CVADIS,NPAR,NSIG,MAXFN,IOPT,cvme,H,G,F,W,IER)
      c     *** Relabel the refined estimates of location
            a1CV = cvme(1)
      c     *** Increment the a1CV counter
            count(c,nn,11) = count(c,nn,11) + 1
            return
            end
            Subroutine CVADIS(NPAR,cvme,F)
      c-----------------------------------------------------------------
      c Purpose:  CVADIS provides the function which is to be minimized
      c           by ZXMIN for the Cramer-von Mises distance measure.
      c           The location parameter is altered to
      c           achieve this minimization.
      c-----------------------------------------------------------------
      c Variables:   NPAR = number of parameters available to alter
      c                n = sample size
      c             cvme = estimates of the parameters being altered
      c                F = value of the function to be minimized
      c                x = array of ordered Pareto variates
      c                c = shape parameter
      c               zi = array of Pareto cdf points
      c              ACV = the squared quantity in the W2 formula
      c             SCVM = the sum of the ACV quantities
      c               W2 = the CVM distance measure
      c-----------------------------------------------------------------
      c Inputs:      NPAR = number of parameters available to alter
      c                n = sample size
      c             cvme = initial estimates (the blu estimates)
      c                x = array of ordered Pareto variates
      c                c = shape parameter
      c-----------------------------------------------------------------
      c Outputs:        F = value of the function at the final estimates
      c             cvme = revised estimates of location and scale;
      c                    these are the CVM minimum distance estimates
      c-----------------------------------------------------------------
      c Calculations:
      c              z(1) = 1 - (1 + [x(1)-a]/b)**(-c)
      c
      c           ACV(1) = [ z(1) - (2i-1)/2n ]**2
```

```
c
c               SCVM = ACV(1) + ACV(2) + ... + ACV(n)
c
c                 WZ = SCVM + 1/12n
c
c-----------------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1     aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1     ,andb,estb,icntb,andab,estaa,estbb,icntab
     1     ,a1CV,b1CV,a1KS,b1KS
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1     ,icntb,icntab
        real cvme(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1     ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,aZAD,bZAD
     1     ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
     1     ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
     1     ,zi(18),SCVM,ACV(18),WZ
        double precision dseed
        SCVM = 0
        do 10 j=1,n
           zi(j) = 1-(1/(1+(x(j)-cvme(1))/bblu))**c
           ACV(j) = (zi(j) - (2*j-1)/(2*real(n)))**2
           SCVM = SCVM + ACV(j)
10      continue
        WZ = SCVM + (1/(12*real(n)))
        F = WZ
        return
        end
        Subroutine CVBMD
c-----------------------------------------------------------------------
c Purpose:  CVBMD generates the minimum distance estimates of
c           the scale parameter based upon minimizing the
c           Cramer-von Mises distance measure defined in
c           subroutine CVBDIS.  CVBMD uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Cramer-von Mises (CVM) distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of CVM distance at the final
c                   parameter estimates
c            cvme = CVM derived minimum distance estimate
c            b1CV = CVM minimum distance scale estimate
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c           count = array of counters used to count the number of
```

```
c                    valid estimate values found
c-----------------------------------------------------------------------
c Inputs:     NPAR = number of parameters altered while minimizing
c             NSIG = number of significant digits required
c            MAXFN = maximum number of function evaluations
c             IOPT = options selector (see IMSL manual on ZXMIN)
c             cvme = initial estimate for the minimization process
c             ablu = blu estimate of location
c             bblu = blu estimate of scale
c-----------------------------------------------------------------------
c Outputs:       F = minimum value of the function being minimized
c             cvme = revised estimate values
c             b1CV = revised MD estimate of scale [b1CV=cvme(1)]
c          H, G, W = vectors defined in IMSL manual on ZXMIN
c              IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c-----------------------------------------------------------------------
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1       ,andb,estb,icntb,andab,estaa,estbb,icntab
     1       ,a1CV,b1CV,a1KS,b1KS
        external CVBDIS
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1       ,icntb,icntab
        real cvme(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1       ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,aZAD,bZAD
     1       ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
     1       ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
        double precision dseed
c       *** Enter the ZXMIN required constants
        NPAR = 1
        NSIG = 3
        MAXFN = 500
        IOPT = 0
c       *** Initialize the cvme value to the blu estimate
        cvme(1) = bblu
c       *** Call ZXMIN to refine the cvme values by minimizing
c       ***      the Cramer-von Mises distance (W2) computed in
c       ***      the subroutine CVBDIS
        call ZXMIN(CVBDIS,NPAR,NSIG,MAXFN,IOPT,cvme,H,G,F,W,IER)
c       *** Relabel the refined estimates of scale
        b1CV = cvme(1)
c       *** Increment the b1CV counter
        count(c,nn,12) = count(c,nn,12) + 1
        return
        end
        Subroutine CVBDIS(NPAR,cvme,F)
c-----------------------------------------------------------------------
c Purpose:  CVBDIS provides the function which is to be minimized
```

```
c           by ZXMIN for the Cramer-von Mises distance measure.
c           The scale parameter is altered to
c           achieve this minimization.
c-----------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
c                n = sample size
c             cvme = estimates of the parameters being altered
c                F = value of the function to be minimized
c                x = array of ordered Pareto variates
c                c = shape parameter
c               zi = array of Pareto cdf points
c              ACV = the squared quantity in the W2 formula
c             SCVM = the sum of the ACV quantities
c               W2 = the CVM distance measure
c-----------------------------------------------------------------
c Inputs:      NPAR = number of parameters available to alter
c                n = sample size
c             cvme = initial estimates (the blu estimates)
c                x = array of ordered Pareto variates
c                c = shape parameter
c-----------------------------------------------------------------
c Outputs:        F = value of the function at the final estimates
c             cvme = revised estimates of scale;
c                    these are the CVM minimum distance estimates
c-----------------------------------------------------------------
c Calculations:
c              z(i) = 1 - (1 + [x(i)-a]/b)**(-c)
c
c           ACV(i) = [ z(i) - (2i-1)/2n ]**2
c
c             SCVM = ACV(1) + ACV(2) + ... + ACV(n)
c
c               W2 = SCVM + 1/12n
c
c-----------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1      aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD,anda,esta,icnt
     1      ,andb,estb,icntb,andab,estaa,estbb,icntab
     1      ,a1CV,b1CV,a1KS,b1KS
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1      ,icntb,icntab
        real cvme(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1      ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,a2AD,b2AD
     1      ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
     1      ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
     1      ,zi(18),SCVM,ACV(18),W2
        double precision dseed
        SCVM = 0
        do 10 j=1,n
           zi(j) = 1-(1/(1+(x(j)-ablu)/cvme(1)))**c
```

```
          ACV(j) = (zi(j) - (2*j-1)/(2*real(n)))**2
          SCVM = SCVM + ACV(j)
10      continue
        W2 = SCVM + (1/(12*real(n)))
        F = W2
        return
        end
        Subroutine KSAMD
c-----------------------------------------------------------------------
c Purpose:  KSAMD generates the minimum distance estimates of
c           location  based upon minimizing the
c           Kolmogorov distance measure defined in subroutine
c           KADIS.  This routine uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Kolmogorov distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of Kolmogorov distance at the final
c                   parameter estimates
c             kse = Kolmogorov derived minimum distance estimates
c            alKS = Kolmogorov minimum distance location estimate
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c           count = array of counters used to count the number of
c                   valid estimate values found
c-----------------------------------------------------------------------
c Inputs:    NPAR = number of parameters altered while minimizing
c            NSIG = number of significant digits required
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c             kse = initial estimates for the minimization process
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c-----------------------------------------------------------------------
c Outputs:     F = minimum value of the function being minimized
c             kse = revised estimate values
c            alKS = revised MD estimate of location [alKS = kse(1)]
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c-----------------------------------------------------------------------
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1      aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1      ,andb,estb,icntb,andab,estaa,estbb,icntab
```

```
     1       ,a1CV,b1CV,a1KS,b1KS
             external KADIS
             integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1       ,icntb,icntab
             real kse(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1       ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,a2AD,b2AD
     1       ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
     1       ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
     1       ,zi(18),SCVM,ACV(18),WZ
             double precision dseed
c            *** Enter the ZXMIN required constants
             NPAR = 1
             NSIG = 3
             MAXFN = 500
             IOPT = 0
c            *** Initialize the kse values to the blu estimates
             kse(1) = ablu
c            *** Call ZXMIN to refine the kse values by minimizing
c            ***      the Kolmogorov distance (KST) computed in the
c            ***      subroutine KADIS
             call ZXMIN(KADIS,NPAR,NSIG,MAXFN,IOPT,kse,H,G,F,W,IER)
c            *** Relabel the refined estimates of location
             a1KS = kse(1)
c            *** Increment the a1KS counter
             count(c,nn,13) = count(c,nn,13) + 1
             return
             end
             Subroutine KADIS(NPAR,kse,F)
c-----------------------------------------------------------------------
c Purpose:  KADIS provides the function which is to be minimized
c           by ZXMIN for the Kolmogorov distance measure.  The
c           location parameter is altered to achieve
c           this minimization.
c-----------------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
c                n = sample size
c              kse = estimates of the parameters being altered
c                F = value of the function to be minimized
c                x = array of ordered Pareto variates
c                c = shape parameter
c               zi = array of Pareto cdf points
c               DP = positive differences between the EDF and cdf
c               DM = negative differences between the EDF and cdf
c            DPLUS = maximum positive difference
c           DMINUS = maximum negative difference
c              KST = Maximum of DPLUS and DMINUS
c-----------------------------------------------------------------------
c Inputs:      NPAR = number of parameters available to alter
c                n = sample size
c              kse = initial estimates (the blu estimates)
c                x = array of ordered Pareto variates
```

```
c                  c = shape parameter
c-----------------------------------------------------------------------
c Outputs:         F = value of the function at the final estimates
c                kse = revised estimates of location;
c                      these are the Kolmogorov minimum distance
c                      estimates
c-----------------------------------------------------------------------
c Calculations:
c               z(i) = 1 - (1 + [x(1)-a]/b)**(-c)
c
c              DP(1) = ABS[ i/n - z(1) ]
c
c              DM(1) = ABS[ z(i) - (i-1)/n ]
c
c-----------------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,nn,count,aZAD,bZAD,anda,esta,icnt
     1       ,andb,estb,icntb,andab,estaa,estbb,icntab
     1       ,a1CV,b1CV,a1KS,b1KS
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1       ,icntb,icntab
        real kse(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1       ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,aZAD,bZAD
     1       ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
     1       ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
     1       ,zi(18),DP(18),DM(18),DPLUS,DMINUS,KST
        double precision dseed
c       *** Calculate the Pareto cdf value [zi(j)] at each point
c       ***      and the differences between the EDF step function
c       ***      and the cdf points
        do 10 j=1,n
           zi(j) = 1-(1/(1+(x(j)-kse(1))/bblu))**c
           DP(j) = ABS(j/real(n) - zi(j))
           DM(j) = ABS(zi(j) - (j-1)/real(n))
10      continue
c       *** Select the maximum of the plus and minus differences
        DPLUS = MAX(DP(1),DP(2),DP(3),DP(4),DP(5),DP(6),DP(7)
     1           ,DP(8),DP(9),DP(10),DP(11),DP(12),DP(13),DP(14)
     1           ,DP(15),DP(16),DP(17),DP(18))
        DMINUS = MAX(DM(1),DM(2),DM(3),DM(4),DM(5),DM(6),DM(7)
     1           ,DM(8),DM(9),DM(10),DM(11),DM(12),DM(13),DM(14)
     1           ,DM(15),DM(16),DM(17),DM(18))
c       *** Select the maximum Kolmogorov distance measure and
c       ***      set F equal to that distance.  F becomes the
c       ***      function which ZXMIN attempts to minimize by
c       ***      altering the values of the location parameter
        KST = MAX(DPLUS,DMINUS)
        F = KST
        return
        end
```

```
      Subroutine KSBMD
c-----------------------------------------------------------------------
c Purpose:  KSBMD generates the minimum distance estimates of
c           scale based upon minimizing the
c           Kolmogorov distance measure defined in subroutine
c           KBDIS.  This routine uses the blu estimates as the
c           starting points for the estimate modifications.
c-----------------------------------------------------------------------
c Variables: NPAR = number of parameters altered by minimizing
c                   the Kolmogorov distance
c            NSIG = number of significant digits for convergence
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c               F = value of Kolmogorov distance at the final
c                   parameter estimates
c             kse = Kolmogorov derived minimum distance estimates
c            b1KS = Kolmogorov minimum distance scale estimate
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c           count = array of counters used to count the number of
c                   valid estimate values found
c-----------------------------------------------------------------------
c Inputs:    NPAR = number of parameters altered while minimizing
c            NSIG = number of significant digits required
c           MAXFN = maximum number of function evaluations
c            IOPT = options selector (see IMSL manual on ZXMIN)
c             kse = initial estimates for the minimization process
c            ablu = blu estimate of location
c            bblu = blu estimate of scale
c-----------------------------------------------------------------------
c Outputs:     F = minimum value of the function being minimized
c             kse = revised estimate values
c            b1KS = revised MD estimate of scale [b1KS = kse(1)]
c         H, G, W = vectors defined in IMSL manual on ZXMIN
c             IER = error parameter (see IMSL manual on ZXMIN)
c-----------------------------------------------------------------------
c Calculate:  no calculations performed in this subroutine
c-----------------------------------------------------------------------
c       *** Variables Declaration:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1       aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD,anda,esta,icnt
     1       ,andb,estb,icntb,andab,estaa,estbb,icntab
     1       ,a1CV,b1CV,a1KS,b1KS
        external KBDIS
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1       ,icntb,icntab
        real kse(1),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1       ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,a2AD,b2AD
     1       ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
```

```
     1      ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
     1      ,z1(18),SCVM,ACV(18),W2
            double precision dseed
c           *** Enter the ZXMIN required constants
            NPAR = 1
            NSIG = 3
            MAXFN = 500
            IOPT = 0
c           *** Initialize the kse values to the blu estimates
            kse(1) = bblu
c           *** Call ZXMIN to refine the kse values by minimizing
c           ***      the Kolmogorov distance (KST) computed in the
c           ***      subroutine KSDIS
            call ZXMIN(KSDIS,NPAR,NSIG,MAXFN,IOPT,kse,H,G,F,W,IER)
c           *** Relabel the refined estimates of location
            b1KS = kse(1)
c           *** Increment the b1KS counter
            count(c,nn,14) = count(c,nn,14) + 1
            return
            end
            Subroutine KSDIS(NPAR,kse,F)
c-----------------------------------------------------------------------
c Purpose:  KSDIS provides the function which is to be minimized
c           by ZXMIN for the Kolmogorov distance measure.  The
c           scale parameter is altered to achieve
c           this minimization.
c-----------------------------------------------------------------------
c Variables:   NPAR = number of parameters available to alter
c                n = sample size
c              kse = estimates of the parameters being altered
c                F = value of the function to be minimized
c                x = array of ordered Pareto variates
c                c = shape parameter
c               z1 = array of Pareto cdf points
c               DP = positive differences between the EDF and cdf
c               DM = negative differences between the EDF and cdf
c            DPLUS = maximum positive difference
c           DMINUS = maximum negative difference
c              KST = Maximum of DPLUS and DMINUS
c-----------------------------------------------------------------------
c Inputs:      NPAR = number of parameters available to alter
c                n = sample size
c              kse = initial estimates (the blu estimates)
c                x = array of ordered Pareto variates
c                c = shape parameter
c-----------------------------------------------------------------------
c Outputs:       F = value of the function at the final estimates
c              kse = revised estimates of scale;
c                    these are the Kolmogorov minimum distance
c                    estimates
c-----------------------------------------------------------------------
```

```
c Calculations:
c              z(i) = 1 - (1 + [x(i)-a]/b)**(-c)
c
c             DP(i) = ABS[ i/n - z(i) ]
c
c             DM(i) = ABS[ z(i) - (i-1)/n ]
c
c-----------------------------------------------------------------------
c       *** Variable Declarations:
        common n,x,c,ablu,bblu,dseed,B,D,Anc,Bnc,BB,aKS,bKS,
     1     aCVM,bCVM,aAD,bAD,nn,count,a2AD,b2AD,anda,esta,icnt
     1     ,andb,estb,icntb,andab,estaa,estbb,icntab
     1     ,a1CV,b1CV,a1KS,b1KS
        integer NPAR,NSIG,MAXFN,IOPT,n,c,nn,count(4,5,14),icnt,hh
     1     ,icntb,icntab
        real kse(i),H(1),G(1),W(3),F,x(18),ablu,bblu,aAD,bAD
     1     ,B(18),D,Anc,Bnc,BB(18),aKS,bKS,aCVM,bCVM,a2AD,b2AD
     1     ,anda(500),esta(500),AD1,andb(500),estb(500),andab(500)
     1     ,estaa(500),estbb(500),a1CV,b1CV,a1KS,b1KS
     1     ,zi(18),DP(18),DM(18),DPLUS,DMINUS,KST
        double precision dseed
c       *** Calculate the Pareto cdf value [zi(j)] at each point
c       ***     and the differences between the EDF step function
c       ***     and the cdf points
        do 10 j=1,n
           zi(j) = 1-(1/(1+(x(j)-ablu)/kse(1)))**c
           DP(j) = ABS(j/real(n) - zi.j),
           DM(j) = ABS(zi(j) - (j-1)/real(n))
10      continue
c       *** Select the maximum of the plus and minus differences
        DPLUS = MAX(DP(1),DP(2),DP(3),DP(4),DP(5),DP(6),DP(7)
     1            ,DP(8),DP(9),DP(10),DP(11),DP(12),DP(13),DP(14)
     1            ,DP(15),DP(16),DP(17),DP(18))
        DMINUS = MAX(DM(1),DM(2),DM(3),DM(4),DM(5),DM(6),DM(7)
     1            ,DM(8),DM(9),DM(10),DM(11),DM(12),DM(13),DM(14)
     1            ,DM(15),DM(16),DM(17),DM(18))
c       *** Select the maximum Kolmogorov distance measure and
c       ***     set F equal to that distance.  F becomes the
c       ***     function which ZXMIN attempts to minimize by
c       ***     altering the values of the location parameter
        KST = MAX(DPLUS,DMINUS)
        F = KST
        return
        end
```

BIBLIOGRAPHY

1. Banks, Jerry and John S. Carson, II. Discrete-Event System Simulation. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1984.

2. Bell, C. B. and others. Signal Detection for Pareto Renewal Processes. Contract N00014-80-C-0208. Technical Report No. 8-82 for the Office of Naval Research, San Diego State University, San Diego, CA, Oct 1982 (AD-A120 972).

3. Berger, J. M. and B. Mandelbrot. "A New Model for Error Clustering in Telephone Circuits," IBM Journal of Research and Development, 7: 224-236 (July 1963).

4. Bertrand, 2Lt David E. Comparison of Estimation Techniques for the Four Parameter Beta Distribution. MS Thesis, AFIT/GOR/MA/81D-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1981.

5. Brunk, H. D. An Introduction to Mathematical Statistics. Lexington, MA: Xerox College Publishing, 1965.

6. Buslenko, N. P. and others. The Monte Carlo Method, the Method of Statistical Trials. New York: Pergamon Press Inc., 1966.

7. Champernowne, D. G. "The Graduation of Income Distributions," Econometrica, 20: 591-615 (1952).

8. Conover, W. J. Practical Nonparametric Statistics. New York: John Wiley & Sons, Inc., 1971.

9. Davis, Henry T. and Michael L. Feldstein. "The Generalized Pareto Law as a Model for Progressively Censored Survival Data," Biometrika, 66: 299-306 (1979).

10. Freeman, Harold. Introduction to Statistical Inference. Reading, MA: Addison-Wesley Publishing Company, Inc., 1963.

11. Freiling, E. C. A Comparison of the Fallout Mass-Size Distributions Calculated by Lognormal and Power-Law Models. San Francisco: U.S. Naval Radiological Defense Laboratory, 1966 (AD-646 019).

12. Fisk, Peter R. "The Graduation of Income Distributions," Econometrica, 29: 171-185 (1961).

13. Hammersley, J. M. and D. C. Handscomb. Monte Carlo Methods. Norwich, Great Britain: Fletcher & Son Ltd, 1967.

14. Harris, Carl M. "The Pareto Distribution as a Queue Service Discipline," Operations Research, 16: 307-313 (January-February 1968).

15. Hastings, N. A. J. and J. B. Peacock. Statistical Distributions. London: Butterworth and Co (Publishers) Ltd, 1975.

16. Johnson, Norman L. and Samuel Kotz. Continuous Univariate Distributions-1. Boston: Houghton Mifflin Company, 1970.

17. Kaminsky, Kenneth S. and Paul I. Nelson. "Best Linear Unbiased Prediction of Order Statistics in Location and Scale Families," Journal of the American Statistical Association, 70: 145-150 (1975).

18. Keffer, 2Lt Jim H. Robust Minimum Distance Estimation of the Three Parameter Lognormal Distribution. MS Thesis, AFIT/GOR/MA/83D-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1983.

19. Koutrouvelis, Ioannis Andreou. Estimation of Asymptotic Pareto Laws and the Tail of a Distribution. Contract N00014-72-C-0508. Technical Report No. 34 for the Office of Naval Research, Arlington, VA, August 1975 (AD-A018 173).

20. Kulldorff, Gunnar and Kerstin Vannman. "Estimation of the Location and Scale Parameters of a Pareto Distribution by Linear Functions of Order Statistics," Journal of the American Statistical Association, 68: 218-227 (1973).

21. Liebelt, Paul B. An Introduction to Optimal Estimation. Reading, MA: Addison-Wesley Publishing Company, 1967.

22. Lindgren, B. W. Statistical Theory. New York: The MacMillan Compa , 1968.

23. Mendenhall, William. Introduction to Linear Models and the Design and Analysis of Experiments. Belmont, CA: Duxbury Press, 1968.

24. Mendenhall, William and Richard L. Scheaffer. Mathematical Statistics With Applications. North Scituate, MA: Duxbury Press, 1973.

25. Mood, A. M. and F. A. Graybill, Introduction to the Theory of Statistics. New York: McGraw Hill Inc., 1963.

26. Parr, William C. "Minimum Distance Estimation: A Bibliography," Communications in Statistics A, 10: 1205-1224 (1981).

27. Parr, William C. and William R. Schucany. Minimum Distance and Robust Estimation. Southern Methodist University, Dallas, TX: Department of Statistics, 1979 (AD-A085 183/2).

28. Parr, William C. and William R. Schucany. "Minimum Distance Estimation and Components of Goodness-of-fit Statistics," Journal of the Royal Statistical Society B, 44: 178-189 (1982).

29. Pigou, A. C. The Economics of Welfare. London: MacMillan and Co., Limited, 1948.

30. Porter, Capt James E. III. Draft copy of Modified Kolmogorov-Smirnov, Anderson-Darling, and Cramer-von Mises Tests for the Pareto Distribution with Unknown Location and Scale Parameters. MS Thesis, AFIT/GSO/MA/85D-6. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December, 1985.

31. Pritsker, A. Alan B. Introduction to Simulation and SLAM II. West Lafayette, IN: Systems Publishing Corporation, 1984.

32. Rao, C. Radhakrishna. Advanced Statistical Methods in Biomedical Research. New York: John Wiley & Sons, Inc., 1952.

33. Runyon, Richard P. and Audrey Haber. Business Statistics. Homewood, IL: Richard D. Irwin, Inc., 1982.

34. Stephens, Michael A. The Anderson-Darling Statistic. Stanford University, Stanford, CA: Department of Statistics, 1979 (AD-A079 807).

35. Stephens, M. A. "EDF Statistics for Goodness of Fit and Some Comparisons," Journal of the American Statistical Association, 69: 730-737 (September 1974).

36. Vannman, Kerstin. "Estimators Based on Order Statistics from a Pareto Distribution," Journal of the American Statistical Association, 71: 704-708 (September 1976).

37. Wine, R. Lowell. Statistics for Scientists and Engineers. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1964.

38. Wolfowitz, J. "Estimation by the Minimum Distance Method," Annals of Mathematical Statistics, 5: 9-23 (1953).

39. Wolfowitz, J. "The Minimum Distance Method," Annals of Mathematical Statistics, 28: 75-88 (1957).

40. Wong, Wing-Yue. On the Property of Dullness of Pareto Distributions. Contract N00014-75-C0455. Technical Report No. 82-16 for the Office of Naval Research, Purdue University, West Lafayette, IN, May 1982 (AD-A119 631).

Vita

Major Dennis J. Charek was born on 21 December 1949 in Cleveland, Ohio. He graduated from Revere High School in Richfield, Ohio in 1967 and attended Kent State University from which he received the degree of Bachelor of Science in Physics in 1971. Upon graduation, he enlisted in the USAF in the accounting and finance specialty area. He received his commission in 1973 through the Officer Training School program. He was first employed as a Space Surveillance Officer at Detachment 7, 14th Missile Warning Squadron, MacDill AFB, FL in 1973. He then went overseas in 1977 to the 13th Missile Warning Squadron, Clear AFS, AK to serve as a Space Surveillance Officer. In 1978, he was called to serve a tour at the Aerospace Defense Command Headquarters at Peterson AFB, CO as the SLBM Radar Space Operations Officer. In 1980, he moved to Strategic Air Command Headquarters, Offutt AFB, NE where he was the Ballistic Missile Early Warning System Radar Manager. During this tour, he earned a Master of Science in Systems Management degree from the University of Southern California. He then moved to the Foreign Technology Division, Wright-Patterson AFB, OH in 1983 to serve as a Radar Development Analyst. He then entered the School of Engineering, Air Force Institute of Technology in 1984.

Permanent Address: 4691 Sunset Drive
Richfield, Ohio 44286

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A163831

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GSO/MA/85D-3	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/MA	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, OH 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. — PROGRAM ELEMENT NO; PROJECT NO.; TASK NO.; WORK UNIT NO.
11 TITLE (Include Security Classification): See Box 19	
12. PERSONAL AUTHOR(S): Dennis J. Charek, Major, USAF	
13a. TYPE OF REPORT: MS Thesis; 13b. TIME COVERED: FROM _____ TO _____; 14. DATE OF REPORT (Yr., Mo., Day): 1985 December; 15. PAGE COUNT: 119	
16. SUPPLEMENTARY NOTATION	

17 COSATI CODES		
FIELD	GROUP	SUB. GR.
12	01	

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Estimates, Probability Distribution Functions, Distributions, Statistical Analysis, Statistical Inference, Statistical Distributions, Monte Carlo Method, Order Statistics

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: A COMPARISON OF ESTIMATION TECHNIQUES FOR THE THREE PARAMETER PARETO DISTRIBUTION

Thesis Advisor: Dr. Albert H. Moore, Professor

Approved for public release: IAW AFR 190-1.
LYNN E. WOLAVER 16 JAN 86
Dean for Research and Professional Development
Air Force Institute of Technology (AFIT)
Wright-Patterson AFB OH 45433

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr. Albert H. Moore, Professor	22b TELEPHONE NUMBER (Include Area Code): 513-255-2915	22c OFFICE SYMBOL: AFIT/MA

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE UNCLASSIFIED

Block 19. Abstract

This thesis compared minimum distance estimation with best linear unbiased estimation to determine which technique provided the most accurate estimates for location and scale parameters as applied to the three parameter Pareto distribution. Two minimum distance estimators were developed for each of the three distance measures used (Kolmogorov, Cramer-von Mises, and Anderson-Darling) resulting in six new estimators. For a given sample size and shape parameter, the location and scale parameters were estimated. Varying combinations of sample sizes (6, 9, 12, 15, 18) and shape parameters (1, 2, 3, 4) were tested.

A Monte Carlo methodology was used to generate the 1000 sample sets of Pareto random variates for each sample size-shape parameter combination with location and scale parameters both set equal to 1. The best linear unbiased estimator and the six minimum distance estimators provided parameter estimates based on each sample set. Finally, the parameter estimates were compared using the mean square error as the evaluation tool. The results of the research indicate that the best linear unbiased estimation technique provided more accurate estimates of location and scale for the three parameter Pareto distribution than did the minimum distance estimation techniques.